Joint Hand Detection and Rotation Estimation Using CNN

Xiaoming Deng, Yinda Zhang, Shuo Yang, Ping Tan, Liang Chang, Ye Yuan, and Hongan Wang, *Member, IEEE*

***Abstract*— Hand detection is essential for many hand related tasks, e.g., recovering hand pose and understanding gesture. However, hand detection in uncontrolled environments is challenging due to the flexibility of wrist joint and cluttered background. We propose a convolutional neural network (CNN), which formulates in-plane rotation explicitly to solve hand detection and rotation estimation jointly. Our network architecture adopts the backbone of faster R-CNN to generate rectangular region proposals and extract local features. The rotation network takes the feature as input and estimates an in-plane rotation which manages to align the hand, if any in the proposal, to the upward direction. A derotation layer is then designed to explicitly rotate the local spatial feature map according to the rotation network and feed aligned feature map for detection. Experiments show that our method outperforms the state-of-the-art detection models on widely-used benchmarks, such as Oxford and Egohands database. Further analysis show that rotation estimation and classification can mutually benefit each other.**

***Index Terms*— Hand detection, rotation estimation, convolutional neural networks.**

I. Introduction

LOCATING human hands and knowing their pose are extremely useful in human-computer interaction and robotics. It helps computers and robots to understand human intentions [1]–[6], and provides a variety of clues, e.g. force, pose, intention, for high level tasks. While generic object detection benchmarks have been refreshing over the last decade, hand detection and pose estimation from a single image, however, is still challenging due to the fact that hand shapes are of great appearance variation under different wrist rotations and articulations of fingers [7], [8].

In this paper, we propose to solve the hand detection problem jointly with in-plane rotation estimation. These two tasks are closely related and could benefit each other. Calibrating training data under different rotation to upright position results in rotation invariant feature, which relieves the burden of the detection/classification model. While in return, detection model can verify if the rotation estimation is reasonable. However, due to the nature of the convolutional neural networks, rotation invariance is more difficult to achieve than translation invariance, which prevents us from an end-to-end optimization. As a result, previous work [9] usually handle transformation estimation and detection separately or in an iterative fashion, which may not achieve a good optima.

To tackle this issue, we design a derotation layer, which explicitly rotates a feature map up to a given angle. This allows us to jointly optimize the network for two tasks simultaneously (refer to Fig. 2 for the network structure). Recently, spatial transformer networks (ST-CNN) [10] also presented a differentiable module to actively spatially transform feature maps with CNN. However, their transformation is learned unsupervised such that could be any arbitrary rotation that can not be interpreted directly (the discussion that ST-CNN may not be an ideal hand detection model are shown in the appendix). Also, the transformation space is typically huge and would require much more data and time to converge. Comparatively, our rotation estimation network is aimed for upright alignment, such that the output can be directly used for related tasks, e.g. hand pose estimation. It is also trained supervised, which is more likely to converge.

The overall pipeline of our system is shown in Fig. 1. Our network is built on the backbone of Faster R-CNN with a derotation layer inserted after the ROI pooling. Taking the local feature after ROI pooling for each of the proposal as input, the rotation estimation network outputs an in-plane rotation, which is taken by the derotation layer to align the local feature map. The detection network then takes the aligned feature map and produces a binary classification to tell if the proposal contains a hand. The overall model can be trained end-to-end and runs efficiently during testing process.

The contributions of this paper are mainly in two aspects. First, we propose, by our knowledge, the first framework that jointly estimates the in-plane hand rotation and performs detection. Experiment shows that we achieve significant better performance than state-of-the-art on widely used benchmarks.

Manuscript received November 25, 2016; revised June 12, 2017, August 23, 2017, and October 19, 2017; accepted November 17, 2017. Date of publication December 4, 2017; date of current version January 23, 2018. This work was supported in part by the National Key R&D Program of China under Grant 2016YFB1001201, in part by the National Natural Science Foundation of China under Grant 61473276, Grant 61402040, Grant 61232013, Grant U1435220, in part by the Canada NSERC Grant 611663 and Grant 611664, and in part by the Distinguished Young Researcher Program, Institute of Software, Chinese Academy of Sciences. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Junsong Yuan. *(Corresponding author: Xiaoming Deng.)*

X. Deng, S. Yang, Y. Yuan, and H. Wang are with the Beijing Key Laboratory of Human Computer Interactions, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China (e-mail: xiaoming@iscas.ac.cn; yangshuo114@mails.ucas.ac.cn; yuanye13@mails.ucas.ac.cn; hongan@iscas.ac.cn).

Y. Zhang is with the Department of Computer Science, Princeton University, Princeton, NJ 08544 USA (e-mail: yindaz@cs.princeton.edu).

P. Tan is with the School of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada (e-mail: pingtan@sfu.ca).

L. Chang is with the College of Information Science and Technology, Beijing Normal University, Beijing 100875, China (e-mail: changliang@bnu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2017.2779600

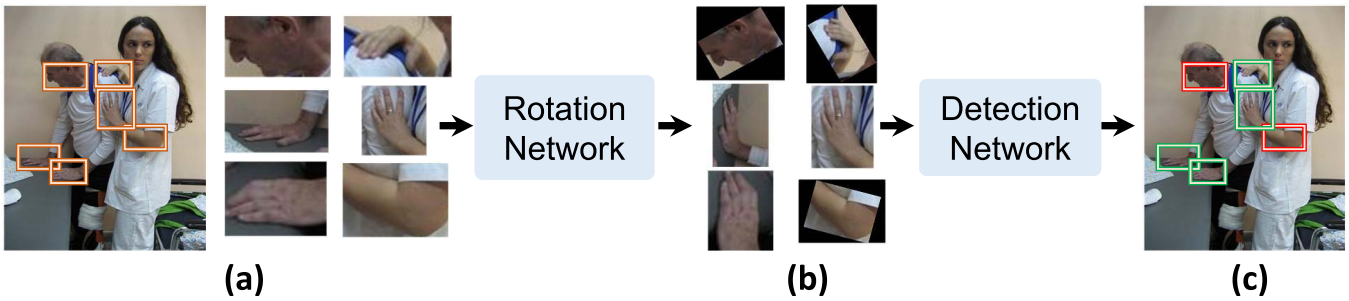


Fig. 1. **Pipeline of our system: Joint hand detection and rotation estimation.** We first generate region proposals from the input image and feed them into the neural network. The in-plane rotation is estimated by the rotation network, and applied back to the input proposal. The aligned data are then fed into the detection network. Thanks to the derotation layer, two tasks are jointly optimized end-to-end.

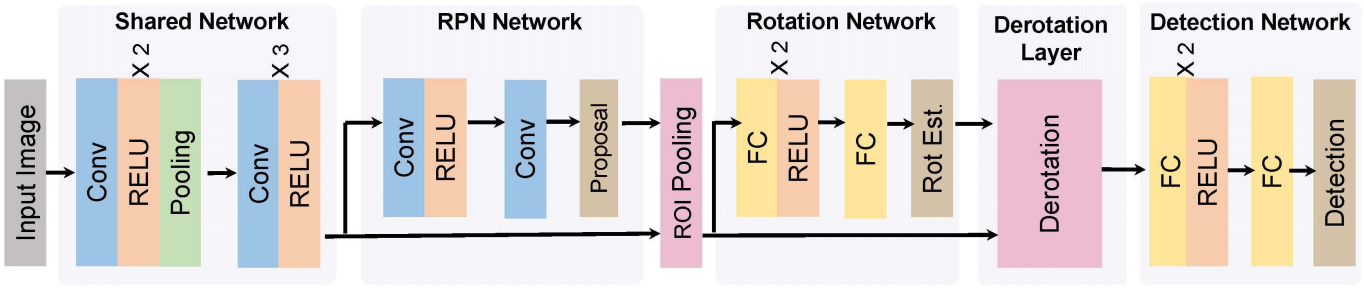


Fig. 2. **Overview of our model with region proposal networks (RPN)**. The network consists of five parts: 1) a shared network for learning features for RPN network, rotation estimation and detection tasks; 2) RPN network to extract hand region proposals; 3) a rotation network for estimating the rotation of a region proposal; 4) a derotation layer for rotating inputted feature maps to a canonical pose; 5) a detection network for classifying the derotated proposal. These modules are jointly used to learn an end-to-end model for simultaneous hand detection and rotation estimation.

Second, we design the derotation layer, which allows end-to-end optimization with two tasks simultaneously. The rotation estimation network is trained with strong supervision, which converges more efficiently.

II. Related Work

Recent hand detection methods from a single image can be classified into four categories:

A. Skin Detection Method

These methods build a skin model with either Gaussian mixture models [11], or using prior knowledge of skin color from face detection [12]. However, these methods often fail to apply to hand detection in general conditions due to the fact that complex illuminations often lead to large variations in skin color and make the skin color modelling problem challenging.

B. Template Based Detection Method

These methods usually learn a hand template or a mixture of deformable part models. They can be implemented by Harr features like Viola and Jones cascade detectors [13], HOG-SVM pipeline [13], mixtures of deformable part models (DPM) [7]. A limitation of these methods is their use of weak features (usually HOG or Harr features). There are also methods that detects human hand as a part of human structure, which uses the human pictorial structure as spatial context for hand position [14]. However, these methods require most parts of human are visible, and occlusion of body parts makes hand detection difficult [15].

C. Per-Pixel Labeling Detection Method

A pixel labeling approach [8] has shown to be quite successful in hand detection in ego-centric videos. In [16], the pixel labeling approach is further extended to a structured image labeling problem. However, these methods require time-consuming pixel-by-pixel scanning for whole image.

D. Detection Method With Pose Estimation

These methods can be classified as two types: 1) first estimate the object pose, and then predict the object label of the image derotated with the object pose; Rowley *et al.* [9] proposed a seminal rotation invariant neural network-based face detection. The system employs multiple networks: the first is a rotation network which processes each input window to determine its orientation, and then uses this information to prepare the window for one or more detector networks. 2) simultaneous pose estimation and detection. He *et al.* [17] proposed a structured formulation to jointly perform object detection and pose estimation. Fidler *et al.* [18] proposed a 3D object detection and viewpoint estimation with a deformable 3D cuboid model. As far as we know, less attention is paid on using convolutional neural networks to jointly model object detection and rotation estimation problems for 2D images.

III. Approach

We present an end-to-end optimized deep learning framework for joint hand detection and rotation estimation with a single image. The overall pipeline is illustrated in Fig. 2.

Our network is built on the backbone of Faster R-CNN with a derotation layer inserted after the ROI pooling. Our method first generates rotation agnostic region proposals. These region proposal can be generated using a typical region proposal network (RPN) from Faster R-CNN or other methods based on deep feature that enables the end-to-end training. Taking the local feature after ROI pooling for each of the proposal as input, the rotation network performs a regression task to estimate an in-plane rotation that could align the local feature map. Then, the local feature map is explicitly rotated according to the result from the rotation network, and passes through the detection network for a confidence score. Since the feature map is supposed to be well aligned, the detection network does not need to handle the alignment and thus performs more reliably. The rotation transformation is done by the derotation layer, which allows back propagation and enables an end-to-end training. Different to ST-CNN [10], both the rotation network and detection network are trained under supervision, therefore the output of the rotation network is guaranteed for the desired data alignment.

A. Proposal Generation

Region proposal generation is important in a typical object detection system, and a variety of category-independent region proposals are proposed including selective search [19], objectness [20], and category independent object proposals [21]. However, there is limited work studying the efficacy of these methods for human hand with extremely articulated shape. Therefore, we conduct a variety of region proposal methods. Besides [19] and [20], we additionally run two competitive methods:

1) Region Proposal Networks: We also use state-of-the-art region proposal networks (RPN) [22] to extract hand proposals. RPN is a fully convolutional network that takes an image as input and outputs a set of object proposals and the corresponding objectness score.

We use state-of-the-art region proposal networks (RPN) [22] to extract hand proposals. RPN is the front end of the Faster R-CNN network that takes an image as input and outputs a set of rectangular shaped object proposals and their corresponding objectiveness scores. RPN can also be easily combined with other components of the network and enable end-to-end training, which is desirable for our purpose.

2) Deep Feature Proposal: We also adopt a standalone discriminative approach to generate hand region proposal. For each testing image, we build a 5-level image pyramid where each layer doubles the resolution of the layer on top. Then, we feed each layer into Alexnet [23] pretrained on ImageNet and extract the conv5 feature. During training, we train 8 SVM binary classifier, with different aspect ratio, to detect hand-like region on the conv5 feature pyramid in a sliding window fashion. This approach allows us to thoroughly search through the translation, scale, and aspect ratio space to guarantee the recall. The thresholds for SVM classifier is calibrated on the validation set as the highest score producing 100% recall, that is, 100 percent of the positive data is covered with at least 0.5 Intersection over Union (IOU). Fig. 5 shows that our method ensures high recall while keeping the number of proposal per image comparable.

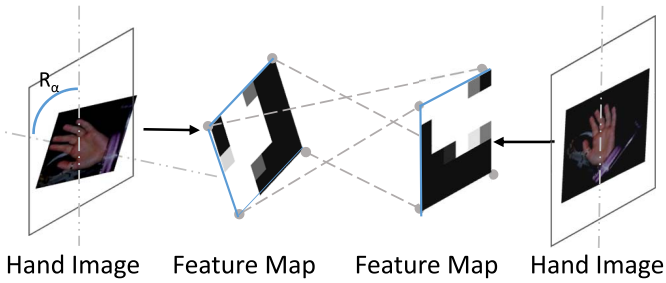


Fig. 3. Illustration of applying derotation transformation to input feature map. The 6×6 feature map is the output of ROI pooling in Fig. 2. The derotation layer aims to warp the input feature map to a canonical hand pose by $\mathbf{R}_\alpha$. In this work, the canonical hand pose is an upright hand pose as shown in the right part of this figure.

B. Rotation Aware Network

In this section, we introduce the rotation aware neural network to decide if a region proposal contains a hand.

1) Network Structure: Our network is built on the backbone of Faster R-CNN with a derotation layer inserted after the ROI pooling (refer to Fig. 2). The network starts with 2 convolution + relu + pooling and 3 convolution + relu to extract features from input image. Built upon this feature, RPN network consists of convolution + relu followed by 1 convolution layers to extract region proposals. Taking the local feature after ROI pooling for each of the proposal, the rotation consists of 3 fully connected layers and estimates the angle to rotate the hand, if any, in the proposal could be aligned to the upward direction. We formulate the rotation estimation problem into a regression problem. Given a rotated hand, the rotation network performs as a regressor and outputs a two dimensional rotation estimation vector $\mathbf{l} = (\cos\alpha, \sin\alpha)$. This representation automatically smooths the jump between 0° and 360° and empirically works better than directly estimating the angle (refer to Table II). In addition, this representation has good geometric meaning. $\mathbf{l}$ means a point on unit circle, Euclidean distance d of two vectors $\mathbf{l}_1$ and $\mathbf{l}_2$, which can be computed with $2\sin\frac{\theta}{2}$, increases as the in-between angle $\theta \in [0, \pi]$ grows. Afterward, a derotation layer rotates the feature from ROI pooling according to the estimated in-plane rotation $\mathbf{l}$ from the rotation network. The rotated feature is then fed into 3 fully connected layers to perform a binary classification, telling if the proposal contains a hand. Since the derotation layer is differentiable, the whole network can be optimized end-to-end, and all tasks can be jointly optimized.

2) Derotation Layer: Derotation layer is a layer which applies a rotation transformation to a feature map during a single forward pass. In our scenario, the input of a derotation layer is the feature map computed from the original image and a in-plane rotation angle predicted from either the rotation network or ground truth, and the output of this layer is the derotated feature map under the given rotation angle, while supposedly under the canonical upright hand pose (refer to Fig. 3).

Specifically, if α is the in-plane rotation angle we want to apply, the derotation transformation is

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}}_{\mathbf{R}_\alpha} \begin{bmatrix} x \\ y \end{bmatrix} \tag{1}$$

where $[x', y']$ is the target canonical coordinates of the regular grid in the output feature map under the canonical upright hand pose, $[x, y]$ is the source coordinates of the regular grid in the input feature map. Note that $[x', y']$ and $[x, y]$ are coordinates relative to the center of feature map.

In our implementations, we use inverse mapping to get the output feature map. In other word, for each pixel $[x', y']$ of the output we get the corresponding position $[x, y]$ in the input feature map. Since $[x, y]$ is often not located on a regular grid, we calculate the feature by averaging the values from its four nearest neighbor locations. We pad zero to $[x, y]$, which is outside of the regular grid.

The back-propagation can be done with a record of the mapping between coordinates between feature map before and after the derotation layer. Updating value on $[x', y']$ is backward propagated to the coordinates from which its value comes, which is in a similar fashion as some pooling layer and ROI layer in [22] and [24].

Denote u^l and u^{l+1} to be the input and output of the derotation layer. Let L be the final loss (refer to Eq.(2)) and $\frac{\partial L}{\partial u^l}$ be the gradient signals of layer l back propagated from the convolutional network. We can calculate the derivative as follows

$$\frac{\partial L}{\partial u^l(x, y)} = \sum_i I(x, y, x_i', y_i') \frac{\partial L}{\partial u^{l+1}(x_i', y_i')}$$

where $[x_i', y_i']$ is the location of pixel i in feature map u^{l+1} (In back-propagation, denote $[x, y]$ to be coordinate located on a regular grid of u^l). The function $I(x, y, x_i', y_i')$ is 1 if the value at $[x, y]$ is propagated to $[x_i', y_i']$, otherwise, $I(x, y, x_i', y_i')$ is 0. Therefore, the partial derivative $\frac{\partial L}{\partial u^{l+1}(x_i', y_i')}$ is accumulated to $\frac{\partial L}{\partial u^l(x,y)}$. The partial derivative $\frac{\partial L}{\partial u^{l+1}(x_i', y_i')}$ is already computed by the backwards function of the layer on top of the derotation layer.

C. Loss Layer

The loss of the network can be defined as follows

$$L = L_{rotation} + \lambda_d L_{detection} + \lambda_R L_{RPN} \tag{2}$$

where $L_{rotation}$ is the rotation network loss (refer to Eq.(3)), $L_{detection}$ is the detection network loss (refer to Eq.(4)), L_{RPN} is the RPN loss as in Faster R-CNN [22]. The terms are weighted by weight parameters λ_d, λ_R, which are set to 1 by default.

1) Rotation Loss: For rotation estimation, we use L_2 loss at the end of rotation network. We get positive hand region proposals and use them to train a network that can do regression on the hand's rotation, formulated as a two-dimensional vector $\mathbf{l} = (\cos\alpha, \sin\alpha) \triangleq (\frac{c}{\sqrt{c^2+s^2}}, \frac{s}{\sqrt{c^2+s^2}})$. Here, c, s are outputs of the final fully connected layer (FC) in rotation network, $\mathbf{l}$ is enforced as a normalized pose vector by normalizing c, s, and thus we can enforce $\mathbf{R}_\alpha$ in Eq.(1) as a rotation matrix. More exactingly, if $\mathbf{l}$ and $\mathbf{l}^*$ are the predicted and ground truth rotation estimation vectors, the rotation loss is

$$L_{rotation}(\mathbf{l}, \mathbf{l}^*) = \frac{1}{m} \sum_i ||\mathbf{l}_i - \mathbf{l}_i^*||_2^2 \tag{3}$$

To deduce the backward algorithm of rotation loss, the partial derivative of $L_{rotation}$ w.r.t $\mathbf{l}_i = (\cos\alpha_i, \sin\alpha_i)$ can be calculated as follows:

$$\frac{\partial L_{rotation}}{\partial \cos\alpha_i} = \frac{2}{m}(\cos\alpha_i - \cos\alpha_i^*)$$

$$\frac{\partial L_{rotation}}{\partial \sin\alpha_i} = \frac{2}{m}(\sin\alpha_i - \sin\alpha_i^*)$$

where i is the index of a positive hand region proposal in a mini-batch, α_i and α_i^* are the estimated and ground truth rotation angle for hand region proposal i, and m is the positive proposal number in a batch. Note that negative training examples do not contribute to back-propagation of the rotation loss. The intuition is that negative training data does not contribute any useful information to the rotation estimation task. Therefore, the negative sample should not change the network weight of the rotation estimation network in any way, which in practice is implemented by locking the gradient to zero during the back-propagation.

We also need to compute the partial derivative of $\mathbf{l}_i = (\cos\alpha_i, \sin\alpha_i)$ w.r.t. c_i, s_i, which can be calculated as follows:

$$\frac{\partial \cos\alpha_i}{\partial c_i} = (c_i^2 + s_i^2)^{-\frac{1}{2}} - c_i^2(c_i^2 + s_i^2)^{-\frac{3}{2}}$$

$$\frac{\partial \cos\alpha_i}{\partial s_i} = -c_i s_i (c_i^2 + s_i^2)^{-\frac{3}{2}}$$

$$\frac{\partial \sin\alpha_i}{\partial c_i} = -c_i s_i (c_i^2 + s_i^2)^{-\frac{3}{2}}$$

$$\frac{\partial \sin\alpha_i}{\partial s_i} = (c_i^2 + s_i^2)^{-\frac{1}{2}} - s_i^2(c_i^2 + s_i^2)^{-\frac{3}{2}}$$

where c_i, s_i are outputs of the final fully connected layer (FC) in rotation network for hand region proposal i in a mini-batch.

2) Detection Loss: For detection task, we use a simple cross-entropy loss at the end of detection network. Denote D^* to be the ground truth object labels, and we use the detection loss as follows

$$L_{detection}(D, D^*) = -\frac{1}{n} \sum_i \sum_j D_i^* \log(D_i) \tag{4}$$

where $D_i = \frac{e^{z_j^i}}{\sum_{j=0}^{1} e^{z_j^i}}$ is the prediction of class j for proposal i given the output z of the final fully connected layer in detection network, n is the training proposal number in a batch.

3) RPN Loss: For RPN network, we use the same loss function as in Faster R-CNN, which consists of objectness classification loss and bounding box regression loss.

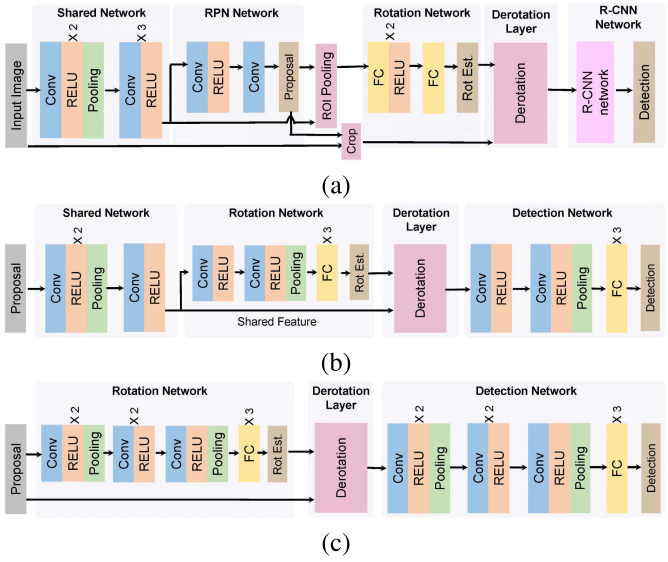


Fig. 4. Network architectures of two comparison pipeline. (a) Our model (RPN, rotate image). (b) Our model (joint). (c) Two-stage CNN. The CNNs before and after derotation are trained one by one. In (b)(c), RELU layer for each fully connected layer is not shown for simplicity.

D. Implementation Details

In this section, we give more details for training with different region proposals.

1) Training With RPN: The network contains three major components - RPN network, rotation estimation network, and detection network. The shared feature map or data flow crossing derotation layer, and therefore require careful training strategy to make sure a good convergence.

We adopt the strategy that first finds good initial for each component and then enables joint optimization. Firstly, the weights of shared network and RPN are initialized by pre-training a model for PASCAL VOC detection benchmark [22]. Secondly, we use the proposals to train rotation network, and then use derotated features after ROI pooling network to train detection network. After the network parameters converge to reasonable good local optima, we enable all the network and optimize in an end-to-end manner for all tasks. During training, the rotation network is randomly initialized by drawing weights from a zero-mean Gaussian distribution with standard deviation 0.01, and the detection network is initialized by by pretraining a model for PASCAL VOC detection benchmark.

We follow the configuration of Faster R-CNN to prepare training data. Each mini-batch randomly samples 256 anchors from a single image, where the ratio of positive and negative anchors is 1:1. A RPN region proposal is considered to be positive if the IOU with a ground truth bounding box is larger than 0.7; negative if the IOU with all ground truth bounding box is smaller than 0.3; discarded otherwise. For negative training data, they do not contribute any gradient during the back propagation from the rotation network.

2) Training With Our Deep Proposals: The network contains two pathways that interact with each other through the derotation layer. To train the model, we adopt a divide and conquer strategy. We first initialize the shared network and the detection network with the model pretrained on ImageNet, and only fine tune on the rotation estimation task. Then, we fix these two networks but enable the detection network, and fine tune for the hand binary classification task. After the network parameters converge to reasonable good local optima, we enable all the network and optimize in a end-to-end manner for both tasks.

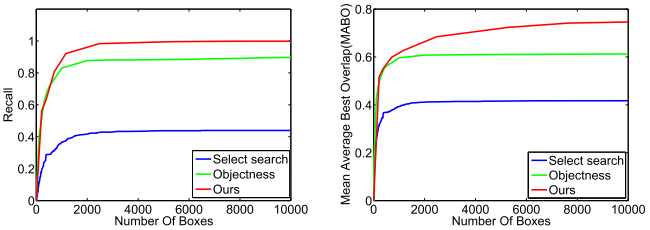


Fig. 5. Trade-off between recall (a) and MABO (b) of the object hypotheses in terms of bounding boxes on the Oxford hand test set.

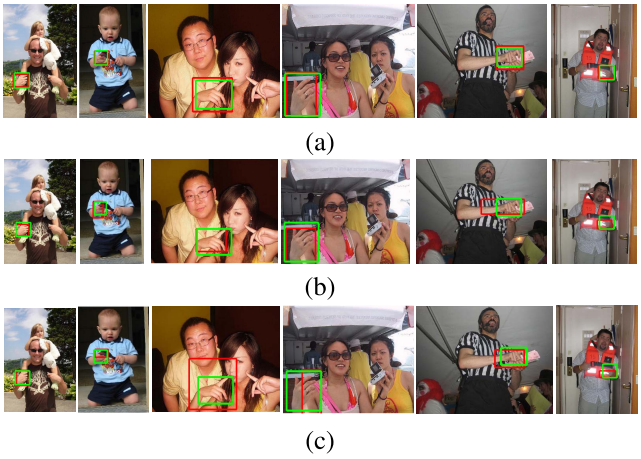


Fig. 6. Comparison between our deep feature proposal generation to the traditional segmentation based algorithms. Examples of locations for objects whose Best Overlap score is around MABO of our method, objectness and selective search. The green boxes are the ground truth. The red boxes are created using our proposal generation method. We only show one proposal for each image to give clear illustration of proposal localization performance. (a) Our deep feature proposal. (b) Objectness. (c) Selective search.

We follow the configuration of R-CNN to prepare training data. We take our deep feature based proposals as the training data. Depending on the IOU with ground truth, a region proposal is considered to be positive if the IOU is larger than 0.5; negative if the IOU is smaller than 0.5; discarded otherwise, which ends up with about 10K positives and 49 million negatives. Since the number of positive and negative data is extremely imbalanced, we use all the positives and randomly sampled 30 million negatives. We also ensure the ratio between positive and negative data in each mini-batch to be 1:1. For the negative data, they do not contribute any gradient during the back propagation from the rotation network.

3) Post-Processing: During testing process, region proposals often overlap with each other, thus we adopt non-maximum suppression (NMS) on region proposals based on their detection score. The IOU threshold for NMS is fixed at 0.3.

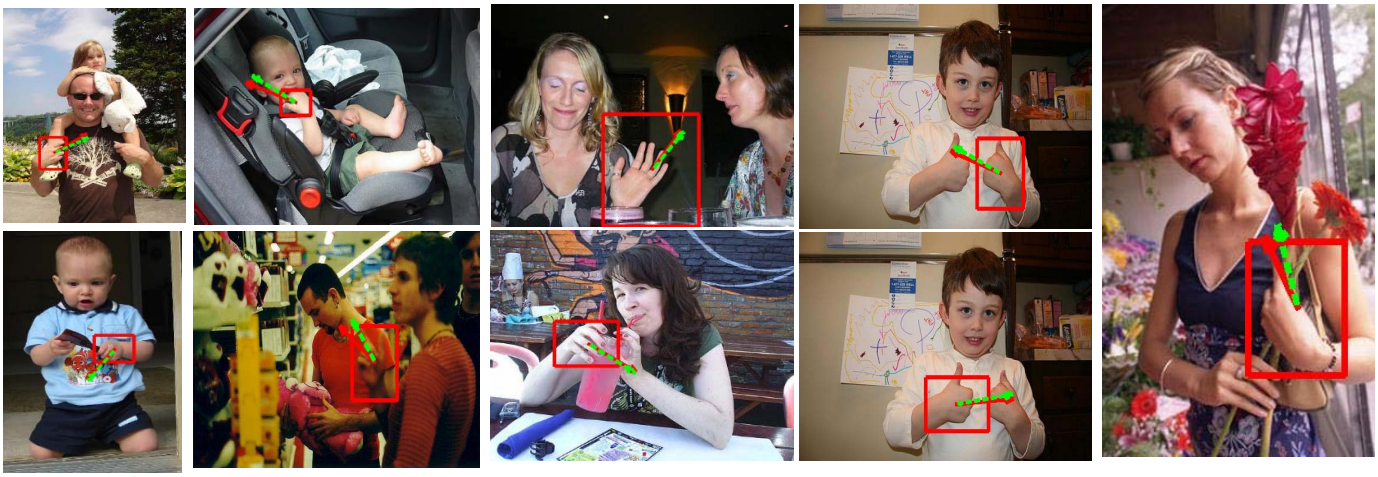


Fig. 7. Examples of hand rotation estimation results on proposals of test images. The red and green arrows indicate the estimated and ground truth rotation angles, respectively. We only show one detected hand for each image in order to give better illustration of rotation estimation results.

IV. Experiments

A. Dataset and Evaluation

The proposed method is evaluated on widely-used Oxford hand dataset [7] and EgoHands dataset [25]. The Oxford hand dataset contains 13050 hands annotated with bounding box and rotation from images collected from various public image datasets. The dataset is considered to be diverse as there is no restriction imposed on the pose or visibility of people, and background environment. Oxford hand dataset has much cluttered background, more viewpoint variations and articulated shape changes than other popular hand dataset such as Signer [26] and VIVA [27]. The EgoHands dataset [25] contains 48 Google Glass videos of complex, first-person interactions between two people, which are also annotated with bounding box and rotation. This dataset is mainly used to recognize hand activities in first-person computer vision.

To demonstrate the performance of the whole system, we evaluate the region proposal generation, the rotation estimation, and final detection performance respectively. For region proposal generation, we measure the percentage of positive data that is covered by any proposal with an IOU larger than 0.5. To further show the localization precision, we also calculate the Mean Average Best Overlap (MABO) [19], which is a standard metric to measure the quantity of the object hypotheses. For rotation estimation, we measure the difference between the estimation and the ground truth. For the detection, we use the typical average precision (AP) and precision-recall curve with a threshold 0.5 on IOU.

We first introduce our model and several alternative architectures as baselines or for ablation study:

1) Our model (RPN): It is based upon Faster R-CNN network, and its details are in Section III;
2) Our model (RPN, rotate image): It is based upon Faster R-CNN and R-CNN. Fig. 4 (a) shows the network architecture. Use RPN to get proposals, estimate rotation with the feature after ROI pooling, extract 300 image proposals, crop them in the input image and resize to 224×224, derotate each image proposal, and then feed to R-CNN for classification;
3) Our model (joint): It is based upon R-CNN network. Fig. 4 (b) shows the network architecture. Use our deep feature based proposal method to get proposals, resize to 224×224, and use an end-to-end training pipeline of the shared network, rotation and detection network;
4) Our model (seperated): It has the same network architecture as our model (joint). The shared 3 convolution layers are kept unchanged, and the rotation and detection networks are trained separately.
5) Two-stage CNN: It is based upon R-CNN network and uses our deep proposal. Fig. 4 (c) shows the network architecture. Use one R-CNN network to regress the 2D rotation for each region proposal, and feed the derotated image proposal to the other R-CNN network for detection;
6) Direct angle regression with CNN: It has the same network architecture as our model (joint) except that

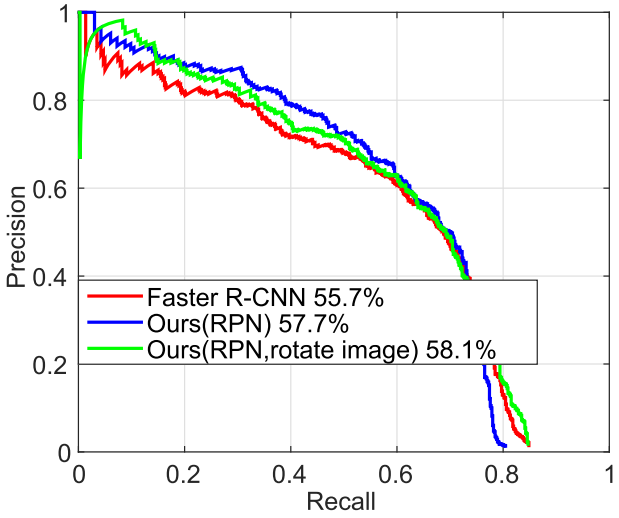


Fig. 8. Precision-Recall curve comparing the Faster R-CNN, our detection model with RPN, and our model with RPN (rotate image) on Oxford hand database.

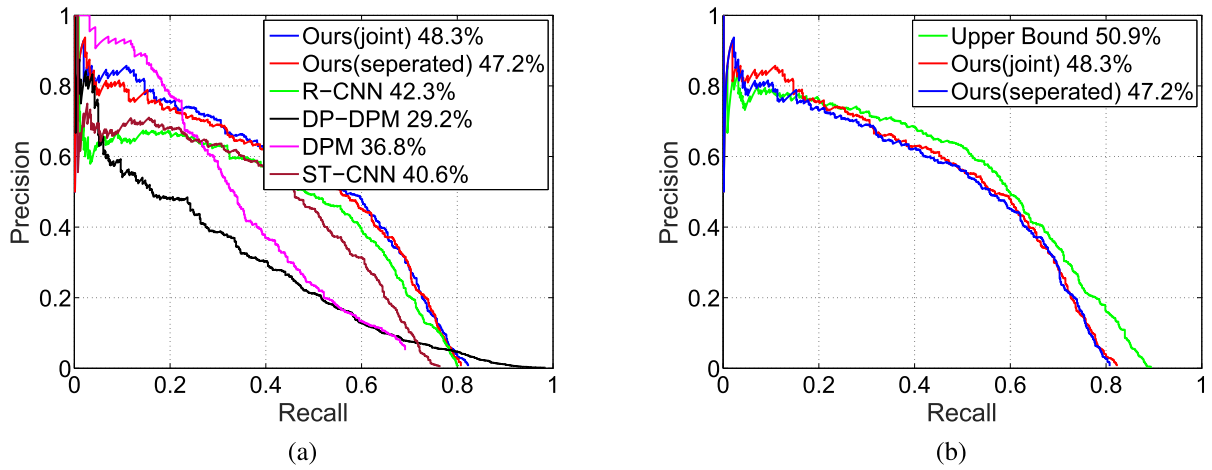


Fig. 9. Precision-Recall curve comparing the baseline and final results. (a) Comparison with baselines. DPM means the results with hand shape detector in [7]. (b) Comparison with detection with ground truth rotation, a performance upper bound.

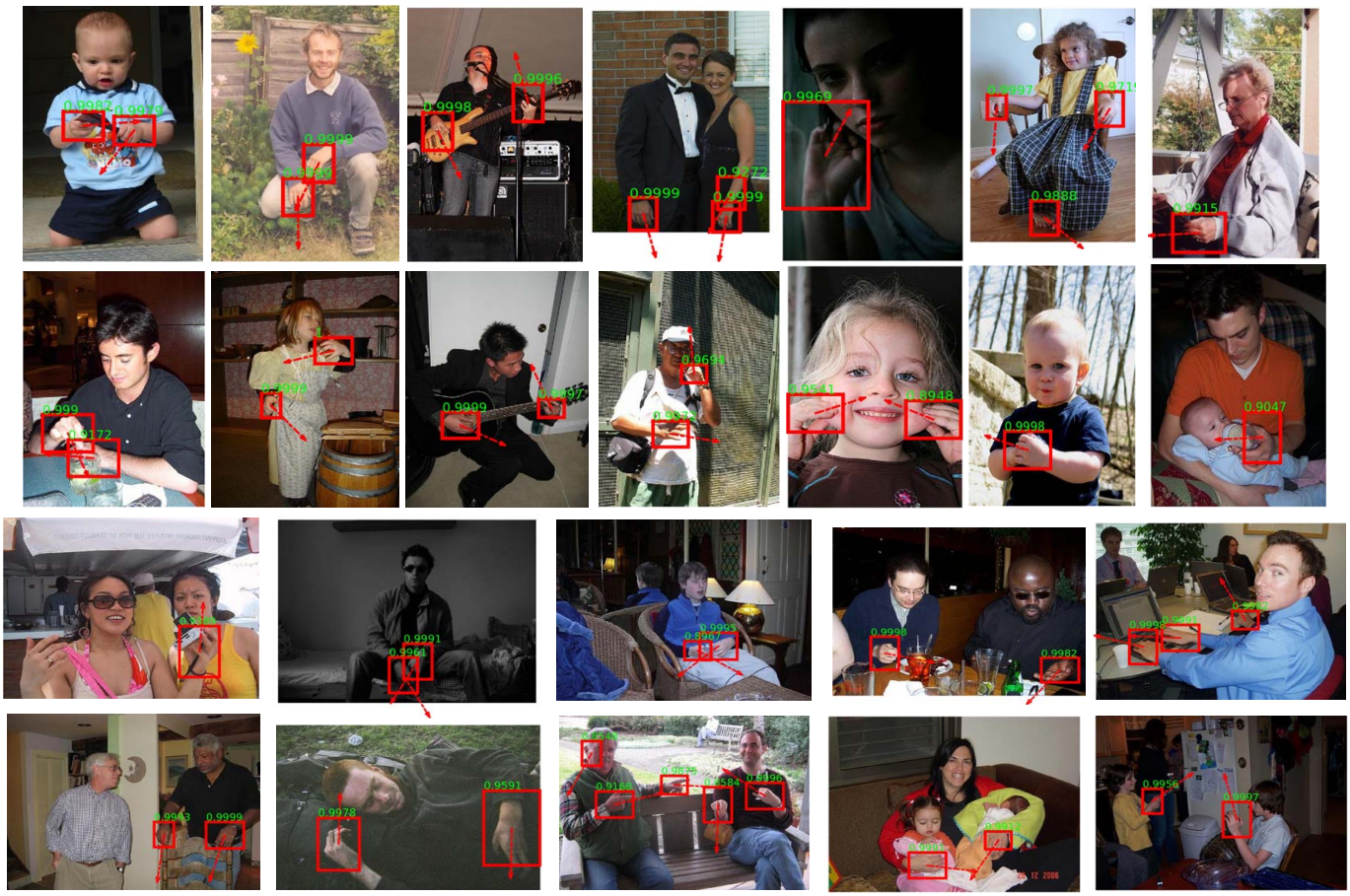


Fig. 10. Examples of high-scoring detection on Oxford hand database. The rotation estimation is illustrated with red arrows.

rotation is estimated with direct angle regression. During the comparison, the vector $\mathbf{l}$ in loss (3) is replaced by the corresponding rotation angle.

B. Performance on Oxford Hand Dataset

1) Region Proposal Generation: Table I show comparisons between RPN [22] and our deep feature proposals to the traditional segmentation based algorithms such as selective search [19] and objectness [20].

In term of the recall, deep feature proposal achieves the best performance, nearly 100%, which is significantly higher than MABO with only about half of the number of proposals (7644 vs. 13000+) used in selective search and objectness. Qualitatively, selective search fails due to the fact that it

TABLE I

PROPOSAL GENERATION PERFORMANCE ON THE OXFORD HAND TEST DATA. #WIN MEANS THE WINDOW NUMBERS

Method	Recall	MABO	#win
Region proposal network (RPN)	95.9%	79.2%	300
Our proposal method	99.9%	74.1%	7644
Our proposal method	100%	76.1%	17489
Selective search	46.1%	41.9%	13771
Objectness	90.2%	61.6%	13000

TABLE II

ROTATION ESTIMATION PERFORMANCE ON THE OXFORD HAND TEST DATA. ROTATION IS CORRECT IF THE DISTANCE IN DEGREE BETWEEN PREDICTION AND GROUND TRUTH IS LESS THAN $\delta_\alpha = 10°, 20°, 30°$. WE COMPARE THE ROTATION ESTIMATION RESULTS ON THE HAND TEST DATA WITH ONLY ROTATION MODEL, AND JOINT ROTATION AND DETECTION MODEL

Method	$\leq 10°$	$\leq 20°$	$\leq 30°$
Our model (RPN)	40.0%	63.8%	75.1%
Our model (joint)	47.8%	70.9%	80.2%
Our model (seperated)	45.6%	70.1%	79.8%
Two-stage CNN	46.1%	70.1%	79.8%
Direct angle regression with CNN	27%	44%	56%

relies much on over-segmentation and may not be suitable for complex scenarios with cluttered background and many connected skin-like regions, while deep feature proposal could take advantage of the discriminative power of the articulated local shape of the hand and generate reliable proposals. Fig. 5 shows that our deep feature proposal ensures high recall while keeping the number of proposal per image comparable. Fig. 6 shows qualitative comparison between our deep feature proposals to selective search and objectness. We can observe that our deep feature proposals have better localization performance. In term of the efficiency, RPN achieves a slightly lower recall than the deep feature proposal but with significantly fewer number of proposals.

2) Rotation Estimation: We first demonstrate that the rotation network can produce reasonable in-plane rotation estimation. Table II shows the performance of the rotation estimation. When RPN is used to extract proposals, the rotation estimation is good. When our deep feature based proposals are used, we can see that the prediction for 45.6% of the data falls in 10 degree around the ground truth, and 70.1% for 20 degree, 79.8% for 30 degree. Examples of hand rotation estimation results on test images are also shown in Fig. 7. We see that our rotation model leads to excellent performance and the rotation estimation results with our deep feature proposal are a little better than those with RPN.

We compare our method with direct angle regression with CNN. During the comparison, the vector $\mathbf{l}$ in loss (3) is replaced by the corresponding rotation angle. We can observe that the two dimensional representation of rotation (refer to results in Table II) leads much better rotation estimation results than direct angle representation.

3) Detection Performance: When RPN is used to extract image region proposals, we compare our model to state-of-the-art approaches such as Faster R-CNN [22]. Faster R-CNN does not explicitly handle rotation. In Fig. 8, we compare the precision-recall using Faster R-CNN and our model with RPN.

TABLE III

AVERAGE PRECISION (AP) ON THE OXFORD HAND TEST DATA

Method	AP
Our model (RPN)	57.7%
Our model (RPN, rotate image)	58.1%
Faster R-CNN	55.7%
Our model (joint)	48.3%
Our model (seperated)	47.3%
Two-stage CNN	46.2%
R-CNN	42.3%
ST-CNN	40.6%
DP-DPM	29.2%
DPM	36.8%

TABLE IV

AVERAGE TIME (SECOND/IMAGE) TO DETECT HANDS. COMPARISON ARE MADE WITH STATE-OF-THE-ARTS DPM-BASED METHOD [7], R-CNN [28], DP-DPM [29], OUR JOINT MODEL AND OUR MODEL WITH RPN. NOTE THAT DPM MEANS THE RUNNING TIME WITH THE AUTHOR'S CODE IN [7]

Method	running time
Our model (RPN)	0.1
Faster R-CNN	0.08
Our model (RPN, rotate image)	1.0
Our model (joint)	8.0
DP-DPM	2.0
R-CNN	9.0
DPM	55.0

Fig. 11. Examples of high-scoring detection on Oxford hand database (outdoor images). The rotation estimation is illustrated with red arrows.

AP of our model with RPN is 2% higher than AP of Faster R-CNN [22]. AP of our model with RPN is 0.4% lower than our model with RPN and rotating image, while our model with RPN is 10 times more efficient (refer to Table IV).

When our deep feature proposals are used, we compare our model to approaches such as R-CNN [28], DPM-based method [7], DP-DPM [29] and ST-CNN [10], the first three of which do not explicitly handle rotation. Fig. 9(a) shows the precision-recall curves, and the number after each algorithm is AP.

Our AP on Oxford hand dataset is 48.3% for our model (joint), which is significantly better (11.5%, 6% higher) than the state of the art [7], in which AP = 36.8% is reached with DPM trained with hand region, and AP = 42.3% is reached with additional data such as hand context and skin color model (we do not use such additional data). Our models,

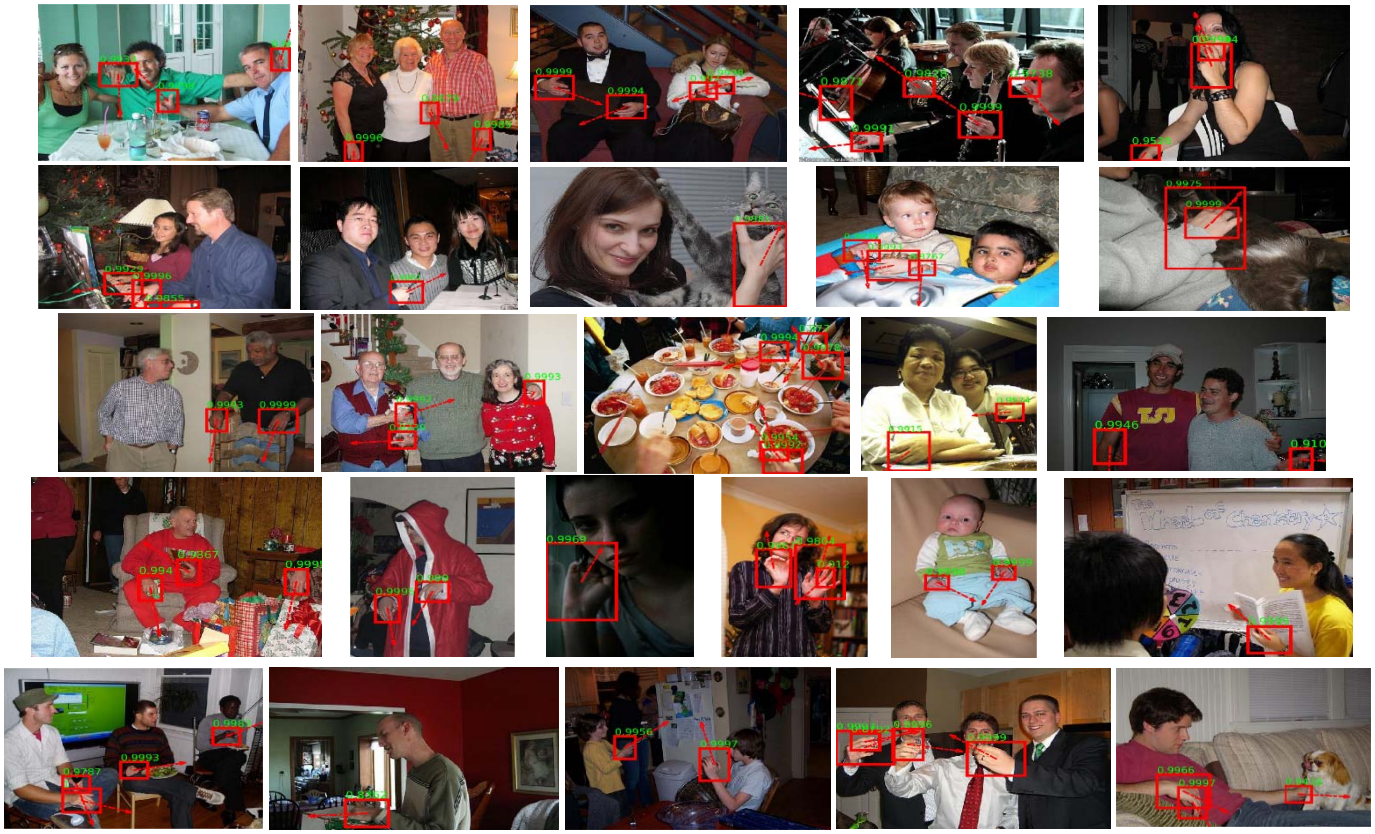


Fig. 12. Examples of high-scoring detection on Oxford hand database (indoor images). The rotation estimation is illustrated with red arrows.

joint or separated, are advantageous over seminal CNN-based methods, AP of our separated model is 5% higher than R-CNN, 6.7% higher than ST-CNN. This demonstrates that data alignment with rotation is very critical for the classification model in the detection network. In Fig. 10, we show some results of our method on test images from Oxford hand dataset, in which both detection bounding boxes and rotation estimation results are shown. The discussion that ST-CNN may not be an ideal hand detection model is shown in the appendix.

We give more experimental results of our hand detector on Oxford hand dataset. Fig. 11 and Fig. 12 are examples of high-scoring detection on Oxford hand database for outdoor and indoor images, respectively. Obviously, our method works well for both outdoor and indoor images, for images with multiple and single hands. We give examples of false alarm detection in Fig. 13, which indicates that skin areas (such as face, arm, foot) are more likely to be misunderstood as hand due to similar skin color, and some non-skin-like regions are also easy to be misclassified. We believe that we can make the hard negative more effective by running a skin area detection [30] and intentionally add negative proposals from the skin area into the training data.

4) Efficiency: We compare the running time with the previous state-of-the-arts method [7], R-CNN [28], DP-DPM [29] in Table IV. Our model with RPN and rotating image is less efficient than our model with RPN, and our model with RPN is more efficient due to the efficient shared convolution feature map, RPN, and ROI pooling. If our deep feature based proposals are extracted, the time efficiency of our method is still superior to that of the method in [7], and it is comparable to that of R-CNN and DP-DPM. The running time with our model (RPN) is about 0.1 seconds per image of 500 $\times$ 400 pixels on a quad-core 2.9GHz PC with Nvidia Titan X, while previous method in [7] takes about 55 seconds per image. Our method is more efficient due to the use of region proposal instead of sliding window, and derotating only once with estimated angle instead of brute force rotating in [7].

C. Model Analysis

1) Is the Model Robust Against Region Proposal?: Our model achieves desirable performance with both deep feature proposal and region proposal network, which shows that our model can jointly work with a variety of region proposal methods.

2) Is The Model Well Optimized?: In order to understand if the model is properly optimized with explicit rotation estimation, we train a detection network with our deep feature proposals and the ground truth rotations. The precision-recall curve is shown in Fig. 9(b). The average precision is 50.9%, which can be considered as a performance upper bound under the current network topology. Again, it shows that aligning data to supervised orientation could great benefit the detection model. Also, our performance is only 2.6% lower than this upper bound, which indicates our system is well optimized.

3) Does Joint Training Help?: Conceptually, it is beneficial to train a network by jointly optimizing over multiple

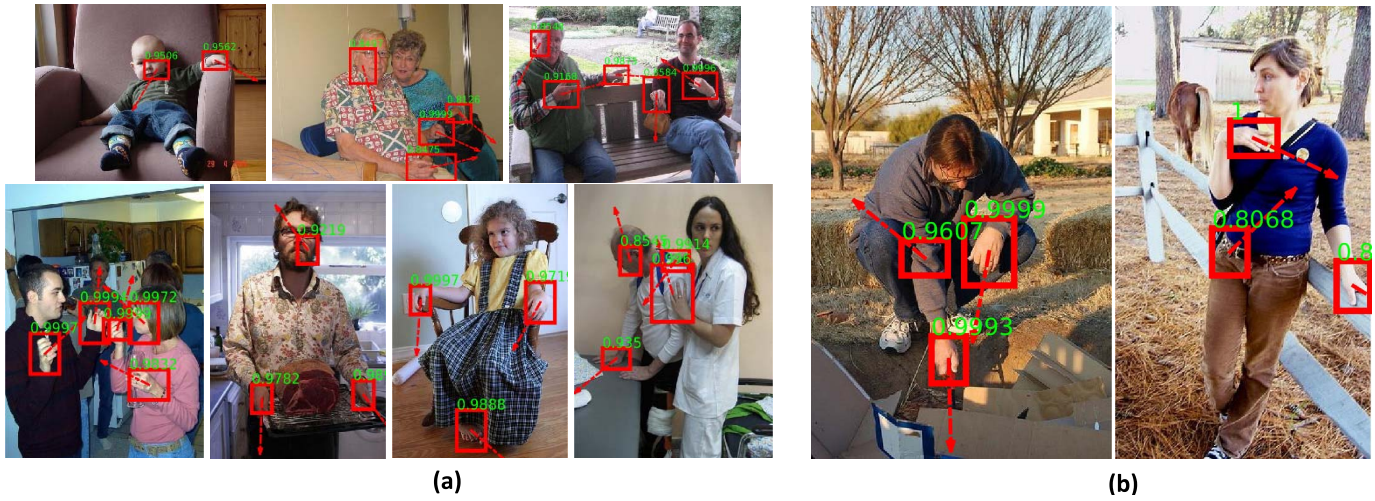


Fig. 13. Examples of false alarm detection on Oxford hand database. (a) false alarm with skin-like region. (b) false alarm with non-skin region.

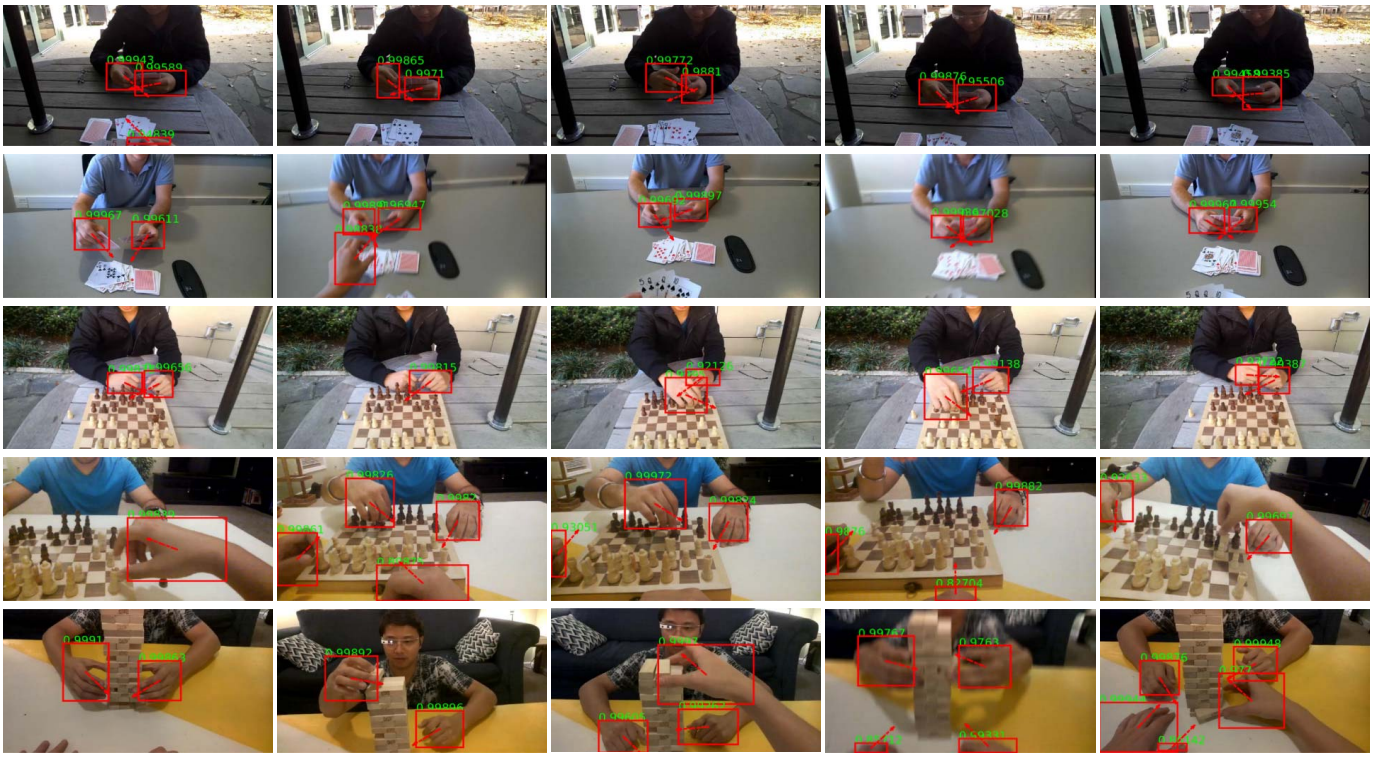


Fig. 14. Examples of high-scoring detection on Egohands database. The rotation estimation is illustrated with red arrows.

related tasks. We investigate this issue here by comparing a model without jointly training to our joint model. To obtain a non-jointly optimized model, our model (seperated), we still follow the divide and conquer fashion of parameter initialization, but allow the rotation network and the detection network to have shared first 3 layers for feature extraction. This results in about 1% drop on rotation estimation (refer to Table II) and 1.1% drop on average precision (refer to Fig. 9(b)).

For the two-stage CNN, we use one CNN for rotation estimation, and the other CNN for detection. This results in about 3% drop on rotation estimation (refer to Table II) and 2.1% drop on average precision (refer to Table III) than our joint model. Overall, we demonstrate that joint training allows multiple tasks to share mutually useful information, and provides a compact and efficient network topology.

4) Derotating Feature or Image Proposal?: We compare our method by derotating feature maps (ours (RPN)) with the method by derotating original image proposals, ours (RPN, rotate image). AP of our model (RPN) is about 0.4% lower than AP of ours (RPN, rotate image). However, our model (RPN) is 10 times efficient than ours (RPN, rotate image), due to the fact that all the proposals share convolution feature map, while each proposal in ours (RPN, rotate image)

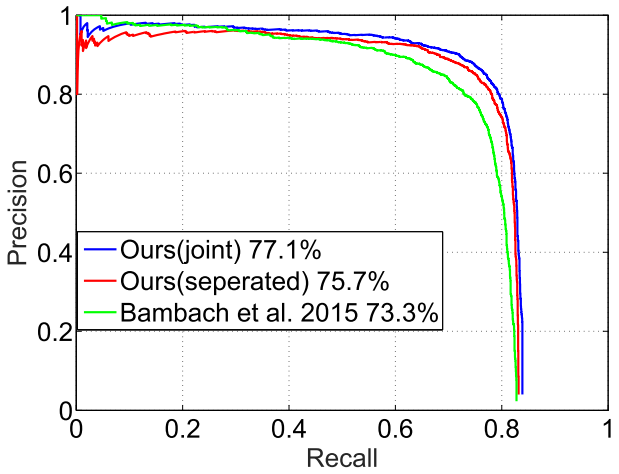


Fig. 15. Precision-Recall curve comparing the baseline and final results on EgoHands dataset [25].

TABLE V
ROTATION ESTIMATION PERFORMANCE ON EGOHANDS DATASET

Method	$\leq 10°$	$\leq 20°$	$\leq 30°$
Our model (joint)	49.0%	76.7%	87.1%
Our model (seperated)	48.6%	76.6%	87.3%

must be warped to the same size and then the convolution features can be calculated, a less efficient pipeline.

5) Does Our Derotation Layer Help Under Different Networks?: We compare hand detection performance under different network architectures. Our model (RPN) is based upon Faster R-CNN, and our model (joint) is based upon R-CNN. AP of our model (RPN) is 2% higher than AP of Faster R-CNN, and AP of our model (joint) is 6% higher than AP of R-CNN. Therefore, the derotation layer is helpful to improve AP for different networks just by adding rotation networks and a derotation layer.

D. Performance on EgoHands Dataset

In order to show the generalization power of our method, we test our pipeline on EgoHands dataset [25]. Fig. 15 shows precision-recall curve comparing the baseline and final results on EgoHands dataset, and the number after each algorithm is the average precision (AP). Our model (seperated) means that the shared 3 convolution layers are kept unchanged, and the rotation and detection networks are trained separately with shared network not tuned, and our model (joint) means that the network is end-to-end trained. Fig. 14 shows examples of high-scoring detection on Egohands database. The rotation estimation performance on Egohands dataset are shown in Table V.

We compared our pipeline with the state-of-the-art detection algorithm in [25]. We implement the state-of-the-art hand detector on this dataset with the network prototxt and Caffemodel provided by [25]. For more rigorous evaluation, we compare detection performance of the method in [25] and our method with the same region proposals, NMS and the other experiment settings. The average precision with our seperated model (AP=75.7%) is higher than the results with baseline (AP=73.3%) (refer to Fig. 15), which indicates that rotation information is helpful to improve the detection results. We then compare the rotation estimation and detection results with separated and joint models. We can see that the rotation estimation results with our joint model is slightly better than the rotation estimation results with separated model. Separated model results in 1.4% drop on average precision than joint model. Therefore, we again demonstrate that joint training allows multiple tasks to share mutually useful information, and provides a compact and efficient network topology.

V. CONCLUSION

Hand detection and pose estimation are important tasks for interaction applications. Previous works mostly solved the problems as separated tasks. In this paper, we explore the feasibility of joint hand detection and rotation estimation with CNN, which is based on our online derotation layer planted in the network. Our experimental results demonstrate that our method is capable of state-of-the-art hand detection on widely-used public benchmarks. The detection network can be extended to use hand context and more sophisticated rotation model.

APPENDIX
PRELIMINARY ANALYSIS ON ST-CNN

We first show that ST-CNN has multiple comparative local optima under different transformation. Take affine transformation as an example, the point-wise transformation layer within ST-CNN is formulated as $x^s = \mathbf{A}_\theta x^t$, where x^t is the target coordinates of the regular grid in the output feature map, x^s is the source coordinates of the input feature map that define the sampling points, and $\mathbf{A}_\theta$ is the affine transformation matrix to optimize.

Suppose $\mathbf{A}_\theta$ after optimization aligns input feature maps into a certain pose. Denote $\mathbf{B}_\beta$ is an arbitrary 2D affine transformation, and obviously $\mathbf{B}_\beta\mathbf{A}_\theta$ can also align feature maps, but in different target poses. As a result, the output feature maps via $\mathbf{A}_\theta$ and $\mathbf{B}_\beta\mathbf{A}_\theta$ are not the same but both aligned. The detection networks trained with two sets of aligned features would have different network weights, but are very likely to have similar detection performance. Therefore, the loss function could reach comparative local minima with either $\mathbf{A}_\theta$ or $\mathbf{B}_\beta\mathbf{A}_\theta$.

We now know that many combinations of transformation parameters and detection weights could result in similar detection performance, i.e. ambiguous rotation estimation and many local minima. The transformation space is typically huge and would require much more data and time to converge. We adopt a supervised approach to get the rotation parameters. Our network will not wonder back and forth between ambiguous transformations, but insists on moving towards the desired pose.

We conduct hand detection experiment with ST-CNN. We add a spatial transformation layer after the input data layer of an Alexnet. Fig. 16 shows hand proposals transformed with affine transformation via ST-CNN. It shows that the hand proposals are not well aligned. In fact, from the result we can see that the ST-CNN fails to learn the transformation that

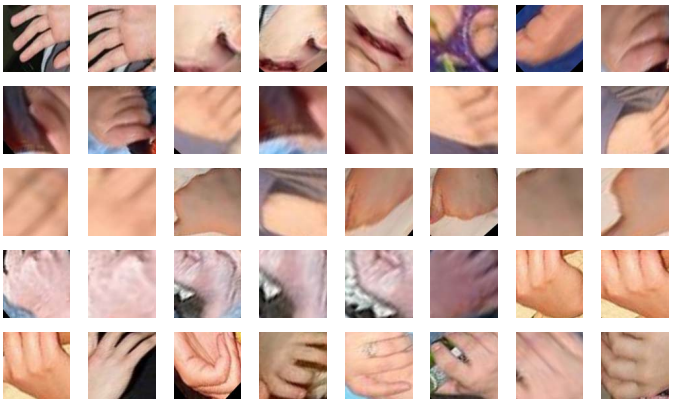


Fig. 16. Hand proposals transformed with affine transformation obtained by ST-CNN.

aligns input proposals, but retreats back to a trivial translation that only captures the major part of the object, i.e. palm region in our case, which is a bad local optima. While the transformed proposals can be still used for the detection network followed, key hand context information is missing (The importance of context for hand and generic object detection is elaborated in [7], [31]). Therefore, the detection performance with ST-CNN could be poor (refer to Fig. 9(a). The performance of ST-CNN is even worse than R-CNN in hand detection task). In summary, for hand detection task, ST-CNN is prone to learning ambiguous transformation, resulting images often miss key context information, which may not be an ideal model for hand detection.

ACKNOWLEDGMENT

The authors are grateful to the editors and reviewers for their careful review and inspiring suggestions. The authors would like to thank Dexin Zuo from ISCAS for help in conducting many comparison experiments. The authors would like to thank Dr. Peng Wang and Dr. Jiewei Cao from University of Queensland, Professor James Landay from Stanford University, and Dr. Sven Bambach from Indiana University Bloomington for helpful discussions. The authors acknowledge the support of NVIDIA with the donation of the GPUs used for this research.

REFERENCES

[1] Y. Yang, C. Fermüller, Y. Li, and Y. Aloimonos, "Grasp type revisited: A modern perspective on a classical feature for vision," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 400–408.

[2] J. Song, F. Pece, G. Sörös, M. Koelle, and O. Hilliges, "Joint estimation of 3D hand position and gestures from monocular video for mobile interaction," in *Proc. 33rd Annu. ACM Conf. Human Factors Comput. Syst.*, 2015, pp. 3657–3660.

[3] G. Rogez, J. S. Supancic, and D. Ramanan, "Understanding everyday hands in action from RGB-D images," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3889–3897.

[4] M. Asad and G. Slabaugh, "Learning marginalization through regression for hand orientation inference," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun./Jul. 2016, pp. 10–18.

[5] H. Liang *et al.*, "Barehanded music: Real-time hand interaction for virtual piano," in *Proc. ACM SIGGRAPH Symp. Interact. 3D Graph. Games*, 2016, pp. 87–94.

[6] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, "Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4207–4215.

[7] A. Mittal, A. Zisserman, and P. H. S. Torr, "Hand detection using multiple proposals," in *Proc. Brit. Mach. Vis. Conf.*, 2011, pp. 1–11.

[8] C. Li and K. M. Kitani, "Pixel-level hand detection in ego-centric videos," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3570–3577.

[9] H. A. Rowley, S. Baluja, and T. Kanade, "Rotation invariant neural network-based face detection," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 1998, pp. 38–44.

[10] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2008–2016.

[11] L. Sigal, S. Sclaroff, and V. Athitsos, "Skin color-based video segmentation under time-varying illumination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 7, pp. 862–877, Jul. 2004.

[12] P. Viola and M. Jones, "Robust real-time object detection," *Int. J. Comput. Vis.*, vol. 4, pp. 51–52, Feb. 2001.

[13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 886–893.

[14] L. Karlinsky, M. Dinerstein, D. Harari, and S. Ullman, "The chains model for detecting parts by their context," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 25–32.

[15] G. Ghiasi, Y. Yang, D. Ramanan, and C. C. Fowlkes, "Parsing occluded people," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2401–2408.

[16] X. Zhu, X. Jia, and K.-Y. K. Wong, "Pixel-level hand detection with shape-aware structured forests," in *Proc. Asian Conf. Comput. Vis.*, 2014, pp. 64–78.

[17] K. He, L. Sigal, and S. Sclaroff, "Parameterizing object detectors in the continuous pose space," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 450–465.

[18] S. Fidler, S. Dickinson, and R. Urtasun, "3D object detection and viewpoint estimation with a deformable 3D cuboid model," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 611–619.

[19] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Apr. 2013.

[20] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2189–2202, Nov. 2012.

[21] I. Endres and D. Hoiem, "Category independent object proposals," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 575–588.

[22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

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

[25] S. Bambach, S. Lee, D. J. Crandall, and C. Yu, "Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1949–1957.

[26] P. Buehler, M. Everingham, D. P. Huttenlocher, and A. Zisserman, "Long term arm and hand tracking for continuous sign language TV broadcasts," in *Proc. Brit. Mach. Vis. Conf.*, 2008, pp. 1105–1114.

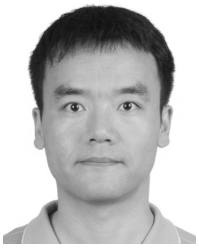
[27] E. Ohn-Bar and M. M. Trivedi, "Beyond just keeping hands on the wheel: Towards visual interpretation of driver hand motion patterns," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst.*, Oct. 2014, pp. 1245–1250.

[28] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

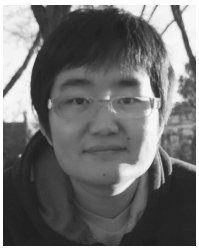
[29] R. Girshick, F. Iandola, T. Darrell, and J. Malik, "Deformable part models are convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 437–446.

[30] C. O. Conaire, N. E. O'Connor, and A. F. Smeaton, "Detector adaptation by maximising agreement between independent data sources," in *Proc. IEEE Int. Workshop Object Tracking Classification Beyond Visible Spectr.*, Jun. 2007, pp. 1–6.

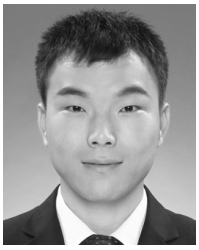
[31] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert, "An empirical study of context in object detection," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1271–1278.



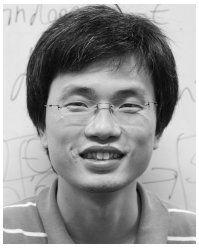
Xiaoming Deng received the B.S. and M.S. degrees from Wuhan University, in 2001 and 2004, respectively, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences (CAS), in 2008. After a two-year post-doctoral at the Institute of Computing Technology, CAS, he joined the Institute of Software, CAS, in 2010, where he is currently an Associate Professor. He was a Research Fellow with the National University of Singapore from 2012 to 2013. His main research topics are in computer vision, and specifically related to 3D reconstruction, human motion tracking and synthesis, and deep learning.



Yinda Zhang received the bachelor's degree from the Department of Automation, Tsinghua University, and the master's degree from the Department of Electrical and Computer Engineering, National University of Singapore, under the supervision of Prof. P. Tan and Prof. S. Yan. He is currently pursuing the Ph.D. degree with Princeton University, advised by Prof. T. Funkhouser. He is currently involved in 3D context model, 3D deep learning, and scene understanding.

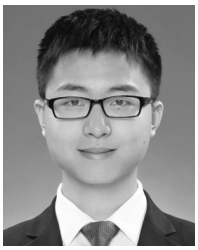


Shuo Yang is currently pursuing the master's degree with the Institute of Software, Chinese Academy of Sciences. His main research interests include computer vision, and specifically related to object detection, hand pose estimation, and face recognition.



Ping Tan received the bachelor's and master's degrees from Shanghai Jiao Tong University, China, in 2000 and 2003, respectively, and the Ph.D. degree in computer science and engineering from the Hong Kong University of Science and Technology in 2007. He was an Associate Professor with the Department of Electrical and Computer Engineering, National University of Singapore. He is currently an Associate Professor with the School of Computing Science, Simon Fraser University, Canada. His research interests include computer vision and computer graphics. He has served as an Editorial Board Member of the *International Journal of Computer Vision*, the *Computer Graphics Forum*, and the *Machine Vision and Applications*.

Liang Chang received the B.S. and M.S. degrees from Wuhan University in 2003 and 2005, respectively, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, in 2008. She joined the College of Information Science and Technology, Beijing Normal University, in 2008, where she is currently an Associate Professor. Her research interests are computer vision and machine learning.



Ye Yuan is currently pursuing the master's degree with the Institute of Software, Chinese Academy of Sciences. His main research interests include computer vision, and specifically related to object detection and face recognition.

Hongan Wang (M'01) received the Ph.D. degree in computer science from the Institute of Software, Chinese Academy of Sciences. He is currently a Professor and the Director of the Beijing Key Laboratory of Human Computer Interactions Laboratory at the Institute of Software, Chinese Academy of Sciences. His research interest is real-time reasoning and intelligent user interaction. He is a member of the IEEE Computer Society and the ACM SIGCHI.