Joint Hand Detection and Rotation Estimation by Using CNN

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

arXiv:1612.02742v1 [cs.CV] 8 Dec 2016

Abstract—Hand detection is essential for many hand related tasks, e.g. parsing hand pose, understanding gesture, which are extremely useful for robotics and human-computer interaction. However, hand detection in uncontrolled environments is challenging due to the flexibility of wrist joint and cluttered background. We propose a deep learning based approach which detects hands and calibrates in-plane rotation under supervision at the same time. To guarantee the recall, we propose a context aware proposal generation algorithm which significantly outperforms the selective search. We then design a convolutional neural network(CNN) which handles object rotation explicitly to jointly solve the object detection and rotation estimation tasks. Experiments show that our method achieves better results than state-of-the-art detection models on widely-used benchmarks such as Oxford and Egohands database. We further show that rotation estimation and classification can mutually benefit each other.

Index Terms—Hand detection, rotation estimation, convoluitonal neural networks.

I. Introduction

Hand detection is fundamental and extremely useful in human-computer interaction and robotics. It helps computers and robots to understand human intentions[1], and provides a variety of clues, e.g. force, pose, intention, for high level tasks. Aside from locating hands in an image, determining the in-plane rotation of the hand is also important as it is usually considered as initialization for other tasks such as hand pose estimation[2] and gesture recognition[3]. While generic object detection benchmarks have been refreshing over the last decade, hand detection 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[4][5].

In this paper, we propose to solve hand detection problem jointly with in-plane rotation estimation. Fig. 1 shows the general pipeline of our system. Inspired by the RCNN [6] framework, we first extract a number of rectangle regions

X.M. Deng, Y. Yuan, S. Yang and H.A. Wang are with Institute of Software, Chinese Academy of Sciences, Beijing, China .
E-mail: xiaoming@iscas.ac.cn, {yuanye13,yangshuo114}@mails.ucas.ac.cn, hongan@iscas.ac.cn
Y.D. Zhang is with Department of Computer Science, Princeton University, Princeton, NJ 08544, USA.
E-mail: yindaz@cs.princeton.edu
P. Tan is with School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada.
E-mail: pingtan@sfu.ca
L. Chang is with College of Information Science and Technology, Beijing Normal University, Beijing, China.
E-mail: changliang@bnu.edu.cn
Manuscript received Nov. 25, 2016.

that are more likely to contain a hand (Fig. 1(a)). Due to the clutter of the image and the articulated shape of the hand, we propose a discriminative proposal generation algorithm, which significantly outperforms the state-of-the-art region proposal methods such as selective search[7] and objectness[8] in term of the recall. The rotation network then estimates the in-plane rotation that align the input hand, if there is, to the upright direction (Fig. 1(b)). The input data are rotated according to this estimated rotation and then fed into the detection network to perform a binary classification (Fig. 1(c)).

Our model is trained jointly with multiple tasks simultaneously, which has been demonstrated to be very successful for many vision tasks. In our case, hand detection and in-plane rotation 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 works [9] usually handle transformation estimation and detection separately or in a iterative fashion, which may not achieve a global optima.

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 (See 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 are not directly interpretable(The discussion that ST-CNN may not be the 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 contributions of this paper are mainly in four 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 benchmark. 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. Third,
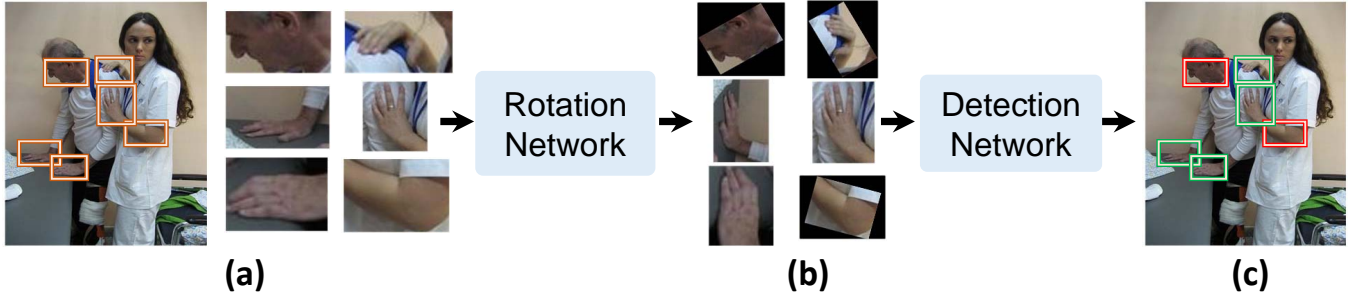
Fig. 1. **Pipeline of our system: Joint hand detection and rotation estimation.** We first generate region proposals from the input image and feed them into the neural network. The in-plane rotation is estimated by the rotation network, and applied back to the input proposal. The aligned data are then fed into the detection network. Thanks to the derotation layer, two tasks are jointly optimized end-to-end.

we propose a hand shape proposal generation algorithm with significantly improved recalls and Mean Average Best Overlap score(MABO) outperforming the state-of-the-art selective search[7] and objectness[8]. Last but not least, our model is much more efficient than previous work. Thanks to the rotation estimation network, we do not need to brute force search for all possible angles and thus reduce the detection time to $1/7$.

II. RELATED WORK

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

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.

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) [4]. 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].

Per-pixel Labeling Detection Method. A pixel labeling approach [5] 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.

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, Baluja, and Kanade[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, Sigal and Sclaroff [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. We first extract proposals for regions that are likely to contain a hand. To particularly take the advantage of the strong local characteristic of hand shape and color, we train a multi-component SVM using features from Alexnet[19] for region proposal rather than simple segmentation based algorithm. The proposals are fed into convolution layers to exact shared features that will be used for both rotation estimation and detection afterward. The rotation network performs a classification task to estimate an in-plane rotation that could align the hand in the input image, if any, to the upward direction. Then, the shared feature is explicitly rotated according to the result from the rotation network, and pass through the detection network for a confidence score. Since the feature 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 enable 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

A variety of category-independent region proposals are proposed including selective search[7], objectness[8], and category independent object proposals[20]. However, due to the cluttered background and articulated shape of the hands, previous region proposals (especially segmentation based) can
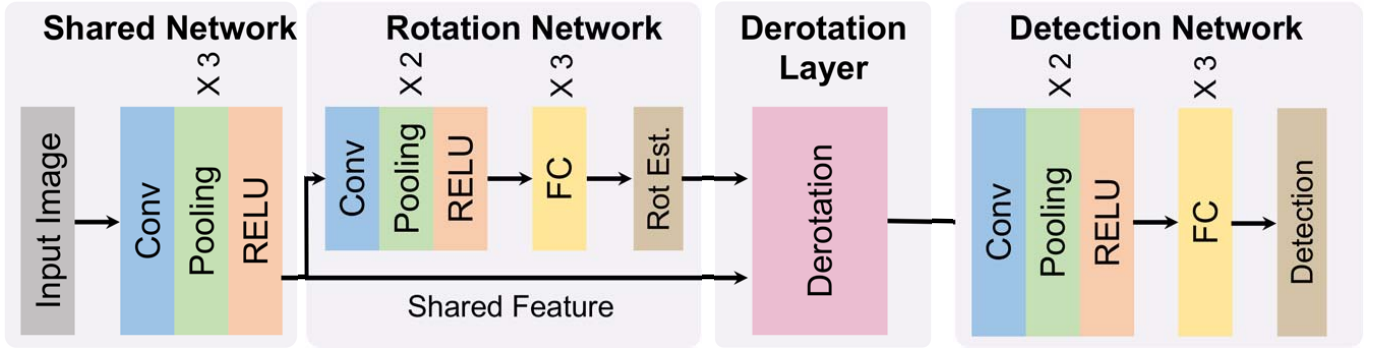
Fig. 2. **Overview of our model.** The network consists of four parts: 1) a shared network for learning features to benefit both rotation estimation and detection tasks; 2) a rotation network for estimating the rotation of a region proposal; 3) a derotation layer for rotating inputted feature maps to a canonical pose; 4) 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.

no longer guarantee recall while keep the number of proposals to be manageable(Table 1 from experiment section shows that selective search and objectness are both not good at hand detection.).

We adopt a discriminative approach to generate hand region proposal. Inspired by deformable parts model(DPM)[21], we cluster the training data to 8 subgroups based on the aspect ratio of the image patches, and train one linear SVM using pooled conv5 layer feature extracted from the Alexnet [19] for each group. We learn the threshold on validation set such that 100 percent of the positive data is covered with at least 0.5 Intersection over Union(IOU). Fig.4 shows that our method ensures significantly higher recall while keeps the number of proposal per image comparable.

B. Rotation Aware Network

In this section, we introduce the rotation aware neural network to decide if a region proposal contains a hand. The detailed network structure is in supplementary material.

1) Network Structure: The network starts with 3 convolution+relu+pooling to extract features from input proposal patches. The resolution of the input feature map is 13×13 due to the strides but still maintains the spatial information. Built upon this feature, the rotation network consists of 2 convolution layers followed by 3 fully connected layer and estimate the angles to rotate such that 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 an rotated hand, the rotation network performs as a regressor and outputs a two dimensional rotation estimation vector $\mathbf{l} = (\cos\alpha, \sin\alpha)$. The estimated $\mathbf{l}$ will then be sent to the derotation layer to rectify the orientation of training patches. Afterward, a derotation layer rotates the feature from the shared network according to the estimated in-plane angle from the rotation network. The rotated feature is then fed into 2 convolution layers and 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 two tasks, rotation estimation and hand detection, can be jointly optimized.
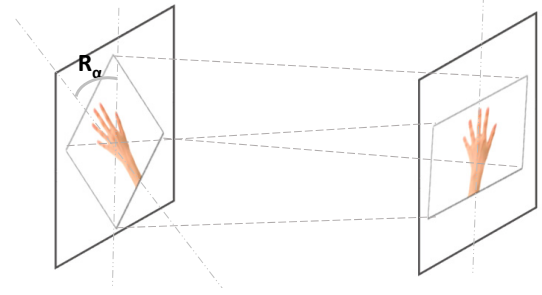

Fig. 3. Illustration of applying derotation transformation to input feature map. The derotation layer aims to warp the input image to a canonical hand pose by $\mathbf{R}_\alpha$. In this work, the canonical hand pose is an upright hand pose as shown in the right part of this figure.

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

$$\left[\begin{array}{c} x' \\ y' \end{array}\right] = \underbrace{\left[\begin{array}{cc} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{array}\right]}_{\mathbf{R}_\alpha} \left[\begin{array}{c} x \\ y \end{array}\right] \qquad (1)$$

where $[x', y']$ is the target 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.

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. An alternative choice is to pad

with feature outside the box, which is not implemented here due to efficiency issue.

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][23].

C. Loss Layer

Our model overall has two losses, the rotation network loss and the detection network loss.

Rotation loss. For rotation estimation, we use L_2 loss. We get ground true hand bounding boxes 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 convolutional layer 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 exactly, if $\mathbf{l}$ and $\mathbf{l}^*$ are the predicted and ground truth rotation estimation vectors, the rotation loss is

$$L_{rotation}(\mathbf{l}, \mathbf{l}^*) = ||\mathbf{l} - \mathbf{l}^*||_2^2 \qquad (2)$$

It is easy to compute the partial derivative of $L_{rotation}$ w.r.t $\mathbf{l} = (\cos\alpha, \sin\alpha)$. To deduce the backward algorithm of rotation loss, we need to compute the partial derivative of $\mathbf{l} = (\cos\alpha, \sin\alpha)$ w.r.t. c, s, which can be calculated as follows:

$$\frac{\partial \cos\alpha}{\partial c} = (c^2+s^2)^{-\frac{1}{2}} - c^2(c^2+s^2)^{-\frac{3}{2}}$$
$$\frac{\partial \cos\alpha}{\partial s} = -cs(c^2+s^2)^{-\frac{3}{2}}$$
$$\frac{\partial \sin\alpha}{\partial c} = -cs(c^2+s^2)^{-\frac{3}{2}}$$
$$\frac{\partial \sin\alpha}{\partial s} = (c^2+s^2)^{-\frac{1}{2}} - s^2(c^2+s^2)^{-\frac{3}{2}}$$

Detection loss. For detection task, we use a simple cross-entropy loss. Denote D^* to be the ground truth object labels, and we use the detection loss as follows

$$L_{detection}(D, D^*) = -\frac{1}{n}\sum_i\sum_j D_i^* \log(D_i) \qquad (3)$$

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

D. Training Schema

The rotation aware 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 rotation 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.

We take only the region proposals from proposal generation section 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.

1) Data Augmentation: Since the pretrained Alexnet on ImageNet does not encode rotation related information, we found that training with the limited number of positive data results in poor generalization capability. We relieve the overfitting by augmenting the size of the training data. We horizontally flip the training image, which allows invariance against left/right hand. We also rotate each positive proposals to 36 times (10 degree once) around the center of the patch, and take each of them as a training data. This dramatically increases the size of training data to 36 times bigger, and our training data eventually contains over 6 million proposals from 4096 images.

2) Hard Negative Mining: Inspired by DPM[21], we perform hard negative mining to suppress confusing false alarms. After training a model from the initial training data, we run the model on the whole training proposals. We add the negatives with high score (≥ 0.4) to the training data and train a model again. Generally speaking, 2-3 times of hard negative mining are sufficient to improve the performance to be almost saturated.

IV. EXPERIMENTS

A. Dataset and Evaluation

The proposed method is evaluated on widely-used Oxford hand dataset[4] and EgoHands dataset[24]. 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[25] and VIVA[26]. The EgoHands dataset [24] 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)[7], 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.

B. Performance on Oxford Hand Dataset

1) Region Proposal Generation: Fig. 5 and Table I show comparisons between our SVM based approach to the traditional segmentation based algorithms such as selective search[7] and objectness[8]. We achieve nearly 100% recall and a significantly higher MABO with only about half the number of proposals (7644 vs. 13000+) used in selective search and objectness. Qualitatively, selective search fails due to the fact that it relies much on over-segmentation and may not be suitable for complex scenarios with cluttered background and many connected skin-like regions, while our method could take advantage of the discriminative power of the articulated local shape of the hand and generate reliable proposals.
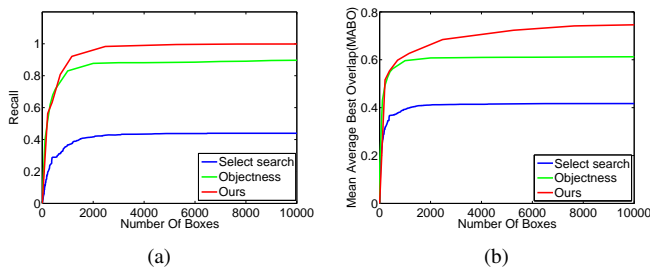

Fig. 4. Trade-off between recall(a) and MABO(b) of the object hypotheses in terms of bounding boxes on the Oxford hand test set.

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

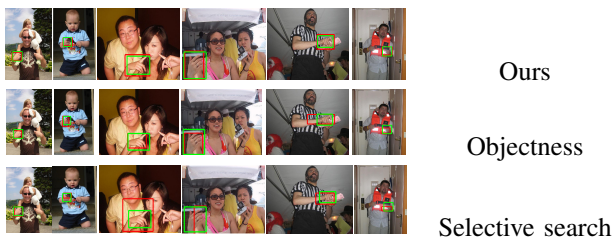| Method	Recall	MABO	#win
selective search	46.1%	41.9%	13771
objectness	90.2%	61.6%	13000
our proposal method	99.9%	74.1%	7644
our proposal method	100%	76.1%	17489

Fig. 5. Comparison between our proposal generation approach 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.

TABLE II
ROTATION ESTIMATION PERFORMANCE ON THE 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	$\leq 10°$	$\leq 20°$	$\leq 30°$
Only rotation model	45.61%	70.13%	79.79%
Joint model	47.84%	70.88%	80.24%

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(Refer to only rotation model). We can see that the prediction for 45.61% of the data falls in 10 degree around the ground truth, and 70.13% for 20 degree, 79.79% for 30 degree. Examples of hand rotation estimation results on test images are also shown in Fig. 6. We see that our rotation model leads to excellent performance.

3) Detection Performance: We compare our model to several state-of-the-art approaches such as R-CNN[6], DPM-based method[4], DP-DPM[27] and ST-CNN[10], the first three of which do not explicitly handle rotation. Fig. **??**(a) shows the precision recall curves, 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. Our average precision 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[4], 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, joint or separated, is advantageous over seminal CNN-based methods, AP of our separated model is 4.9% higher than R-CNN, 6.6% higher than ST-CNN. This demonstrates that data alignment with rotation is very critical for the classification model in the detection network. In Fig. 8, we show some results of our method on test image from Oxford hand dataset, in which both detection bounding boxes and rotation estimation results are shown. The discussion that ST-CNN may not be the ideal hand detection model is shown in the appendix.

We give more experimental results of our hand detector on Oxford hand dataset. Fig. 9 and Fig.10 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.11, 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[28] and intentionally add negative proposals from the skin area into the training data.
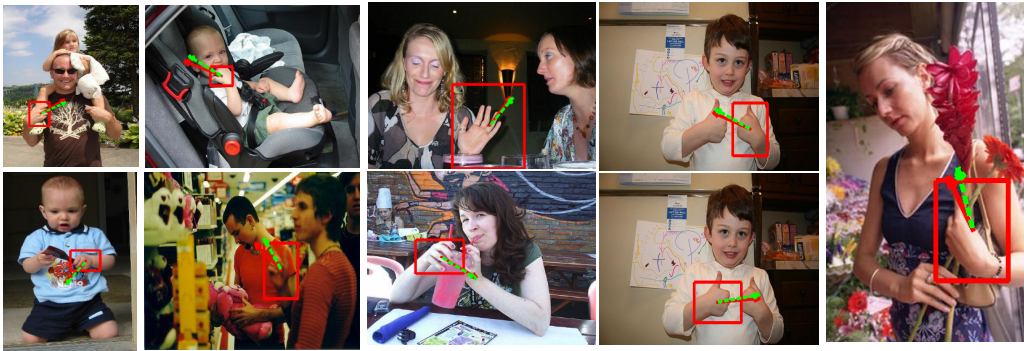
Fig. 6. Examples of hand rotation estimation results on proposals of test images. The red and green arrows indicate the estimated and ground truth rotation angles, respectively.
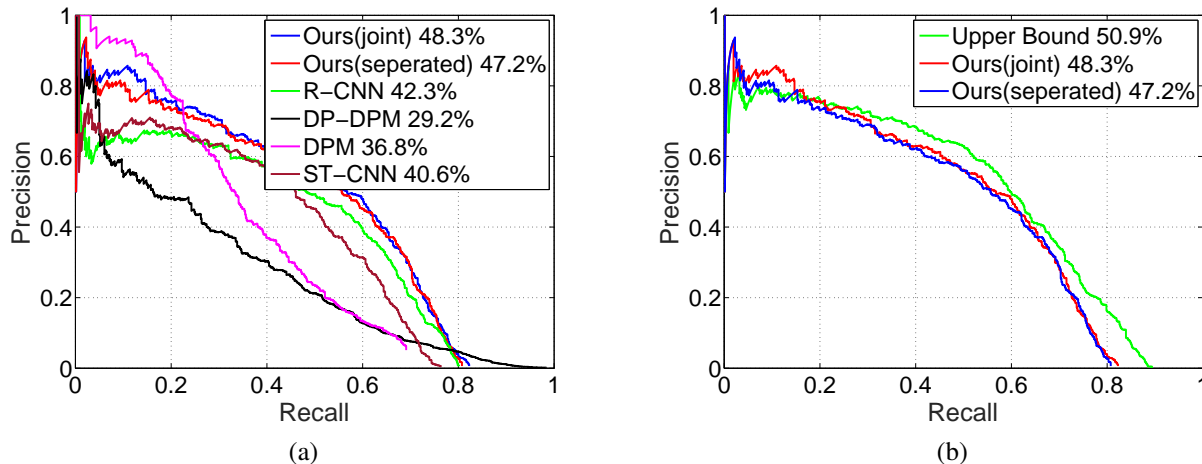

| (a) | (b) |

Fig. 7. Precision-Recall curve comparing the baseline and final results. (a) Comparison with baselines. DPM means the results with hand shape detector in [4]. (b) Comparison with detection with ground truth rotation, a performance upper bound.

TABLE III
AVERAGE TIME (SECOND/IMAGE) TO DETECT HANDS. COMPARISON ARE MADE WITH STATE-OF-THE-ARTS DPM-BASED METHOD[4], R-CNN[6], DP-DPM[27], AND OUR JOINT MODEL. OUR METHOD IS SUPERIOR TO [4] IN RUNNING TIME.

	DPM	R-CNN	DP-DPM	Joint model
running time	55	9	2	8

4) Efficiency: We compare the running time with the previous state-of-the-arts method[4], R-CNN[6], DP-DPM[27] in Table III. The time efficiency of our method is superior to that of the method in[4], and it is comparable to that of R-CNN and DP-DPM. The running time is about 8 seconds per image of 500×400 pixels on a quad-core 2.9GHz PC with Nvidia Titan X, while previous method in [4] 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 [4]. We believe that our method can be more time efficient by leveraging more advanced region proposal method such as region proposal networks[23] and sharing convolutional feature maps of an image for all proposals by using pooling methods such as ROI pooling[22].

C. Model Analysis

1) 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 the ground truth rotation. The precision-recall curve is shown in Fig. 7(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.

2) Does Joint Training Help?: Conceptually, it is beneficial to train a network by jointly optimizing over multiple 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, 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 2% drop on average precision(Refer to Fig. 7(b)) and about 1% drop on rotation estimation(Refer to Table II). Overall, we demonstrate that joint training allows multiple tasks to share mutually useful information, and provides a compact and efficient network topology.

Fig. 8. Examples of high-scoring detection on Oxford hand database. The rotation estimation is illustrated with red arrows.

Fig. 9. Examples of high-scoring detection on Oxford hand database(outdoor images). The rotation estimation is illustrated with red arrows.

D. Performance on EgoHands dataset

In order to show the generalization power of our method, we test our pipeline on EgoHands dataset[24]. Fig.13 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. 12 shows examples of high-scoring detection on Egohands database. The rotation estimation performance on Egohands dataset are shown in Tab.IV.

We compared our pipeline with the state-of-the-art detection algorithm in [24]. We implement the state-of-the-art hand detector on this dataset with the network prototxt and Caffemodel provided by [24]. For more rigorous evaluation, we compare detection performance of the method in [24] and our method with the same region proposals, NMS and the other experiment setting. The average precision with our seperated model (AP:75.7%) is higher than the results with baseline (AP:73.3%)(Refer to Fig.13), which indicates that rotation information is helpful to improve the detection results. We then compare the rotation estimation and detection results
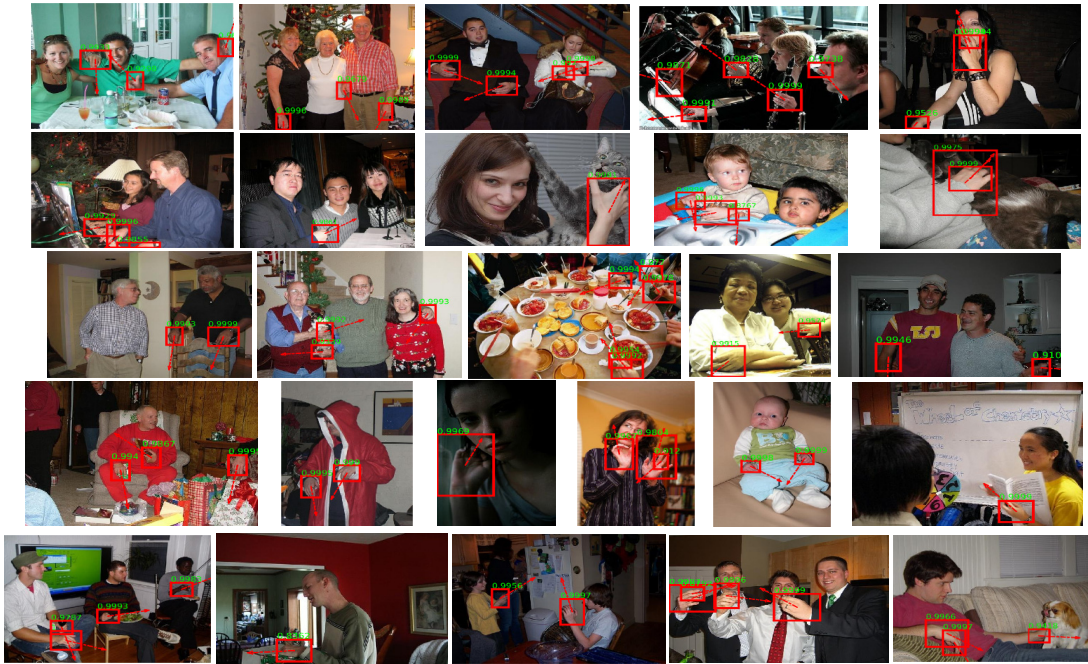
Fig. 10. Examples of high-scoring detection on Oxford hand database(indoor images). The rotation estimation is illustrated with red arrows.

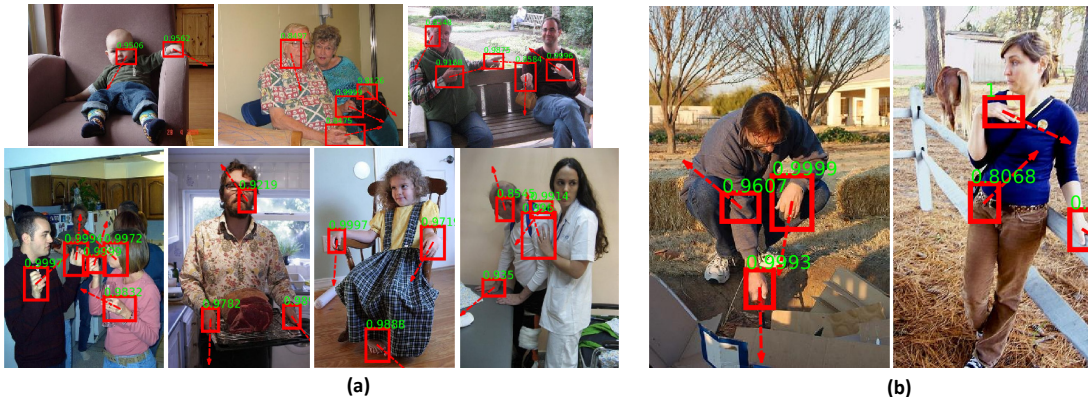**(a)**   **(b)**

Fig. 11. Examples of false alarm detection on Oxford hand database. (a) false alarm with skin-like region. (b) false alarm with non-skin region.
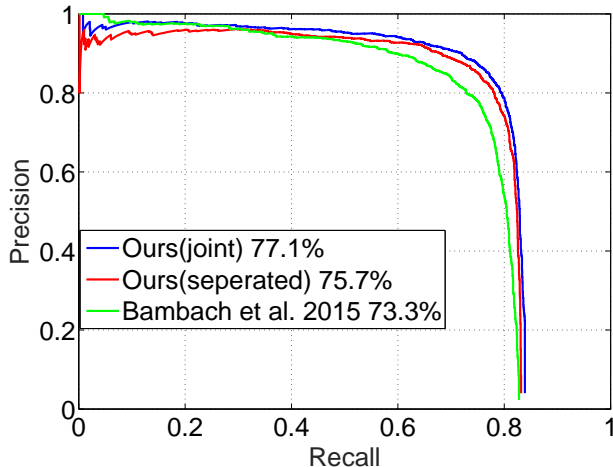

Fig. 13. Precision-Recall curve comparing the baseline and final results on EgoHands dataset[24].

TABLE IV
ROTATION ESTIMATION PERFORMANCE ON EGOHANDS DATASET.

Method	$\leq 10°$	$\leq 20°$	$\leq 30°$
Only rotation model	48.63%	76.56%	87.26%
Joint model	49.01%	76.68%	87.09%

with separated and joint models. We can see that the rotation estimation results with our joint model is slightly better than the results with only rotation 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 problem as separated tasks. In this paper, we explore the
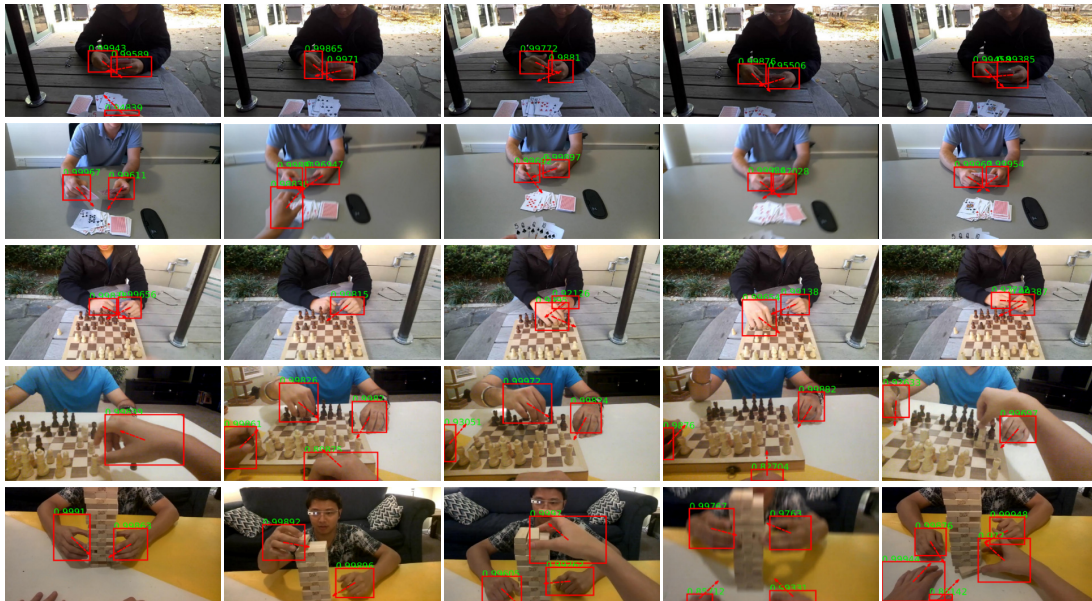
Fig. 12. Examples of high-scoring detection on Egohands database. The rotation estimation is illustrated with red arrows.

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.

ACKNOWLEDGMENT

REFERENCES

[1] Y. Yang, C. Fermller, Y. Li, and Y. Aloimonos, "Grasp type revisited: A modern perspective on a classical feature for vision," in *Proceedings of Computer Vision and Pattern Recognition*. IEEE Press, 2015, pp. 400–408.

[2] R. Wang and J. Popović, "Real-time hand-tracking with a color glove," *ACM transactions on graphics*, vol. 28, no. 3, p. 63, 2009.

[3] R. Wang, S. Paris, and J. Popović, "6d hands: markerless hand-tracking for computer aided design," in *Proceedings of the 24th annual ACM Symposium on User Interface Software and Technology*. ACM Press, 2011, pp. 549–558.

[4] A. Mittal, A. Zisserman, and P. Torr, "Hand detection using multiple proposals." in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2011, pp. 1–11.

[5] C. Li and K. Kitani, "Pixel-level hand detection in ego-centric videos," in *Proceedings of Computer Vision and Pattern Recognition*. IEEE Press, 2013, pp. 3570–3577.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of Computer Vision and Pattern Recognition*. IEEE Press, 2014, pp. 580–587.

[7] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[8] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2189–2202, 2012.

[9] H. Rowley, S. Baluja, and T. Kanade, "Rotation invariant neural network-based face detection," in *Proceedings of Computer Vision and Pattern Recognition*. IEEE Press, 1998, pp. 38–44.

[10] M. Jaderberg, K. Simonyan, and A. Zisserman, "Spatial transformer networks," in *Processing of Advances in Neural Information Processing Systems*. NIPS Press, 2015, pp. 2008–2016.

[11] L. Sigal, S. Sclaroff, and V. Athitsos, "Skin color-based video segmentation under time-varying illumination," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 7, pp. 862–877, 2004.

[12] P. Viola and M. Jones, "Robust real-time object detection," *International Journal of Computer Vision*, vol. 4, pp. 51–52, 2001.

[13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of Computer Vision and Pattern Recognition*. IEEE Press, 2005, pp. 886–893.

[14] L. Karlinsky, M. Dinerstein, D. Harari, and S. Ullman, "The chains model for detecting parts by their context," in *Proceedings of Computer Vision and Pattern Recognition*. IEEE Press, 2010, pp. 25–32.

[15] G. Ghiasi, Y. Yang, D. Ramanan, and C. Fowlkes, "Parsing occluded people," in *Proceedings of Computer Vision and Pattern Recognition*. IEEE Press, 2014, pp. 2401–2408.

[16] X. Zhu, X. Jia, and K. Wong, "Pixel-level hand detection with shape-aware structured forests," in *Processing of Asian Conference on Computer Vision*. Springer Press, 2014, pp. 64–78.

[17] K. He, L. Sigal, and S. Sclaroff, "Parameterizing object detectors in the continuous pose space," in *Proceedings of European Conference on Computer Vision*. Springer Press, 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 *Proceedings of Advances in Neural Information Processing Systems*. NIPS Press, 2012, pp. 611–619.

[19] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of Advances in Neural Information Processing Systems*. NIPS Press, 2012, pp. 1097–1105.

[20] I. Endres and D. Hoiem, "Category independent object proposals," in *Proceedings of European Conference on Computer Vision*, 2010, pp. 575–588.

[21] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[22] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*. IEEE Press, 2015, pp. 1440–1448.

[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Proceedings of Advances in Neural Information Processing Systems*, pp. 91–99, 2015.

[24] S. Bambach, S. Lee, D. Crandall, and C. Yu, "Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions," in *Proceedings of IEEE International Conference on Computer Vision*. IEEE Press, 2015, pp. 1949–1957.

[25] P. Buehler, M. Everingham, D. Huttenlocher, and A. Zisserman, "Long term arm and hand tracking for continuous sign language tv broadcasts," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2008, pp. 1105–1114.

[26] E. Ohn-Bar and M. Trivedi, "Beyond just keeping hands on the wheel: Towards visual interpretation of driver hand motion patterns," in *Proceedings of 17th International IEEE Conference on Intelligent Transportation Systems*. IEEE Press, 2014, pp. 1245–1250.

[27] R. Girshick, F. Iandola, T. Darrell, and J. Malik, "Deformable part models are convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Press, 2015, pp. 437–446.

[28] C. Conaire, N. O'Connor, and A. Smeaton, "Detector adaptation by maximising agreement between independent data sources," in *Proceedings of IEEE International Workshop on Object Tracking and Classification Beyond the Visible Spectrum*. IEEE Press, 2007, pp. 1–6.

[29] S. Divvala, D. Hoiem, J. Hays, A. Efros, and M. Hebert, "An empirical study of context in object detection," in *Proceedings of Computer Vision and Pattern Recognition*. IEEE Press, 2009, pp. 1271–1278.

APPENDIX A
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.14 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 align input proposals, but retreat back to a trivial translation that only captures the major part of the object, i.e. palm region
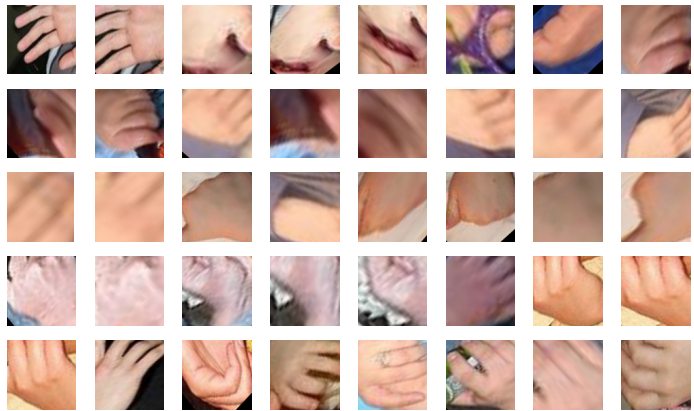

Fig. 14. Hand proposals transformed with affine transformation obtained by ST-CNN.

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 [4][29]). Therefore, the detection performance with ST-CNN could be poor(Please refer to Fig. 7(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 learn ambiguous transformation, resulting images often miss key context information, which may not be the ideal model for hand detection.