YOLSE: Egocentric Fingertip Detection from Single RGB Images

Wenbin Wu, Chenyang Li, Zhuo Cheng, Xin Zhang* and Lianwen Jin
School of Electronic and Information Engineering
South China University of Technology
Guangzhou, P. R. China
eexinzhang@scut.edu.cn

Abstract

With the development of wearable device and augmented reality (AR), the human device interaction in egocentric vision, especially the hand gesture based interaction, has attracted lots of attention among computer vision researchers. In this paper, we build a new dataset named EgoGesture and propose a heatmap-based solution for fingertip detection. Firstly, we discuss the dataset collection detail and as well the comprehensive analysis of this dataset, which shows that the dataset covers substantial data samples in various environments and dynamic hand shapes. Furthermore, we propose a heatmap-based FCN (Fully Convolution Network) named YOLSE (You Only Look what You Should See) for fingertip detection in the egocentric vision from single RGB image. The fingermap is the proposed new probabilistic representation for the multiple fingertip detection, which not only shows the location of fingertip but also indicates whether the fingertip is visible. Comparing with state-of-the-art fingertip detection algorithms, our framework performs the best with limited dependence on the hand detection result. In our experiments, we achieve the fingertip detection error at about 3.69 pixels in 640px $\times$ 480px video frame and the average forward time of the YOLSE is about 15.15 ms.

1. Introduction

Recently, the egocentric hand based interaction, is attracting more and more attention because of the rapid development of the smart wearable devices with camera, such as Google Glass, Microsoft HoloLens and so on. It is natural and easy to use our fingertips to interact in both real world and virtual immersive environment. The location of fingertips and their trajectories play important roles in various HCI applications because they can represent various of interactive instructions, for example, pointing, selecting and pressing. Since this reason, we focus on the fast and accurate fingertip detection from single RGB image in the egocentric vision.

Despite impressive improvements of the Convolution Neural Networks (CNN) based object detection [11, 18, 19], it is still a difficult task to detect multiple fingertips from single RGB images because of complex background, illumination changes, low resolution, fast hand-finger movements, small fingertip size, etc. Additionally, the fingertip is comparably small with limited distinguishable features. Both RGB and depth data are usually used to estimate the 3D hand pose. However, the portable commercial RGB-D camera provides noisy depth data in the indoor environment and the computational load limits its applications on the real-time interaction. Recently, a two-stage Faster R-CNN [19] based single fingertip detection method [8] has been proposed and an egocentric input system is also presented. This method can only detect one fingertip of a special gesture and the detection accuracy can be influenced by the hand detection performance.

In this paper, we establish an egocentric dataset called EgoGesture including 16 hand gestures 59,111 frames with hand and fingertip labeled. With detailed analysis and comparison, we believe that our dataset is of diversity and valuable as a benchmark dataset for the fingertip and hand related research in the egocentric vision. We further propose a heatmap-based fully convolutional network (FCN), named YOLSE (You Only Look what You Should See), for a various number of fingertips detection from single RGB image. In specific, we propose the fingermap as the probabilistic representation of multiple fingertip detection result, which contains the information about both the fingertip location and visibility. In our experiments, the proposed YOLSE reaches the best fingertip detection results comparing with state-of-the-art algorithms. The average multiple fingertip detection error is 3.69 pixels and real-time performance is about 66 frames per second.

2. Related work

We will review related work from three perspectives: pose estimation, fingertip detection and hand-related

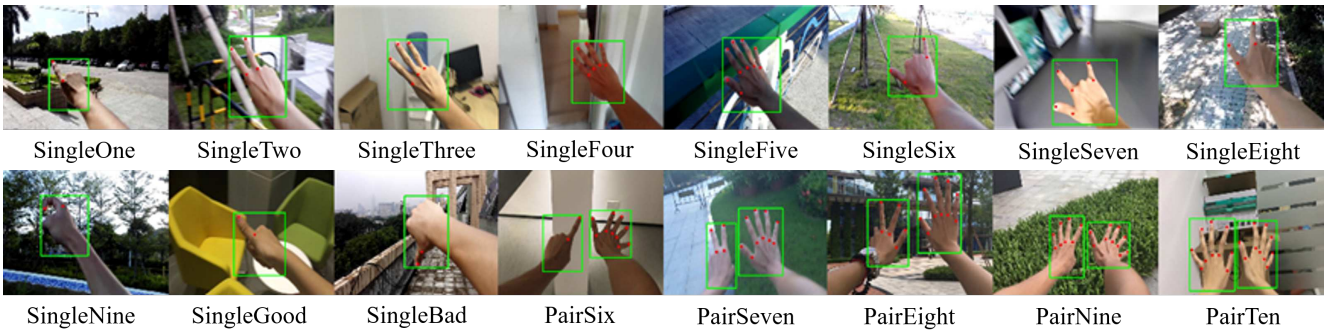


Figure 1. Some representative samples of each gesture.

datasets. Since human body and hand are both articulated non-rigid tree-like structures and the human pose estimation mainly uses RGB images, we include those methods into discussion.

Pose estimation. The CNN-based human pose estimation methods have produced nice results and tested on MPII Human Pose benchmark dataset. Two main strategies have been proposed in this field: direct regression and heatmap regression. Toshev *et al.* [26] proposed a CNN to directly locate the 2D Cartesian coordinates of body joints from color image input. Because of low accuracy caused by the flexible body motion, self-occlusion and weak model expansibility, researchers propose to use the heatmap regression approaches to statistically determine the joint position, like in Tomas *et al.* [16] and Chu *et al.* [4]. Inspired by the heatmap regression approach, we employ the similar statistical map to represent the fingertip detection result. In terms of the hand pose estimation, most methods aim at recovering all the joint positions using depth data [10, 14, 21, 23] or multi-view inputs [6]. Several deep learning based methods have achieved promising results by proposing the multi-resolution image pyramid CNN [25], error feedback loop CNN [15], and hierarchical tree-like structured CNN [13]. For the natural interaction purpose, the commercial depth camera is not suitable due to noisy data and high computational cost. Our research purpose is to locate the fingertip from single RGB images for the interaction, not the full hand structure.

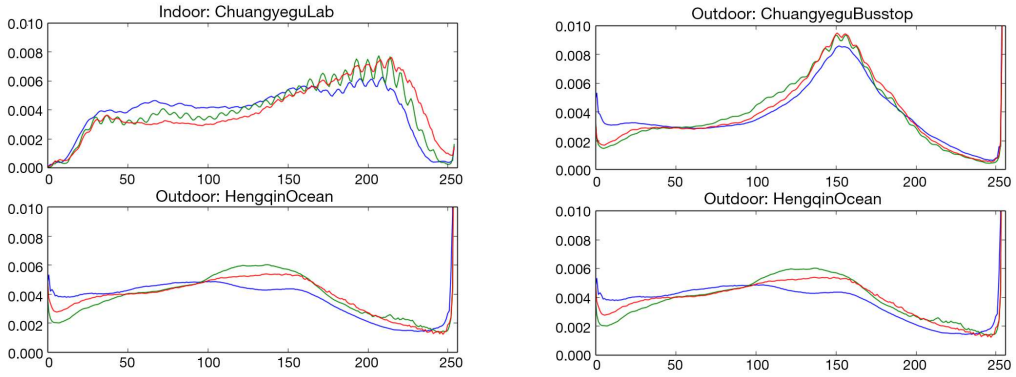
Fingertip detection. Fingertip detection has long been an attractive task in computer vision and HCI. A method to detect fingertip by using skin filters and image cropping is proposed in [17]. Kang *et al.* [9] detects fingertip by applying skin color segmentation and contour extraction. However, these approaches based on skin color usually fail in complex environments and skin-like background. More recent work turn towards the CNN-based approach. [12] proposes a cascaded CNN pipeline for fingertip detection by applying attention-based hand detection firstly. In [8], a two-stage Faster R-CNN based hand detection and dual-target fingertip detection framework is proposed and they design an air writing system for the egocentric vision. Nevertheless, their approach only detects one fingertip in only one gesture, which is difficult to expand. Also the detection accuracy can still be improved for the real world application because their hand detection accuracy has direct and important impact on the fingertip detection.

Related datasets. There are some datasets for hand pose estimation collecting RGB-D images with depth sensor [20, 24, 25, 28], however, these datasets are not designed for egocentric vision. Georgia Tech Egocentric Vision Repository [1] provides a list of datasets on the egocentric vision for the egocentric action recognition, object detection, video summarization, handled object segmentation and so on. However, there is no dataset for fingertip detection and hand gesture interaction in the list of datasets. In [5], a first-person hand action benchmark dataset with RGB-D videos and 3D hand pose annotations is presented to recognize first-person hand actions interacting with 3D objects. A RGB-D dataset with fingertip labeled is established in [22] but regrettably designed for third-person perspective. The latest related dataset, called EgoFinger [8], contains egocentric RGB videos of pointing gesture in different scenarios but it only has one specific gesture. We believe our dataset is diverse and representative as a benchmark dataset for the fingertip and hand related research in the egocentric vision.

3. Dataset: EgoGesture

To recognize various hand gestures and detect corresponding fingertips in egocentric vision using the deep learning method, we need large representative training dataset. So we establish a dataset called **EgoGesture**[1], containing egocentric view based RGB images of various gestures and we manually label the data set by providing the position of the hand, fingertip and some key joints of every gesture. We analyze EgoGesture dataset attributes to show its diversity.

[1]The EgoGesture dataset can be downloaded from http://www.hcii-lab.net/data/



(a) An indoor scene and an outdoor scene (b) Two outdoor scene

Figure 2. Color distribution of different scenes.

The dataset contains 59,111 RGB images in the egocentric vision with 16 different hand gestures (11 single hand gestures and 5 double hand gestures) . We collected the data under different complex backgrounds and illumination conditions. Figure 1 shows some representative samples of each gesture from the dataset.

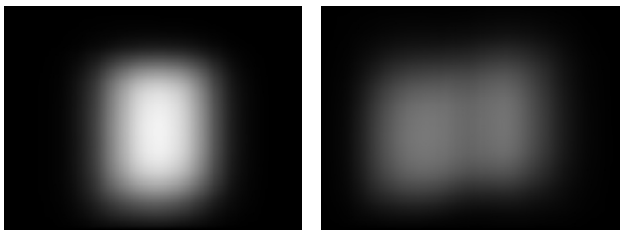
3.1. Dataset details

Considering the rationality and diversity of the EgoGesture dataset, we collected the data under the following conditions: complex backgrounds, illumination change, different user hands and directions, skin-like backgrounds, etc. Furthermore, to meet the requirements of interacting with the device naturally and simply, we design 16 common hand gestures, 11 gestures with single hand: SingleOne (3374 frames), SingleTwo (3763 frames), SingleThree (3768 frames), SingleFour (3767 frames), SingleFive (3755 frames), SingleSix (3757 frames), SingleSeven (3773 frames), SingleEight (3380 frames), SingleNine (3769 frames), SinleBad (3761 frames) and SingleGood (3769 frames); 5 gestures with both hands: PairSix (3681 frames), PairSeven (3707 frames), PairEight (3653 frames), PairNine (3653 frames) and PairTen (3536 frames). We collected samples with not only single hand but also both hands because we believe that the two hands situation will be more and more common as the single hand situation with the development of the wearable camera. To make the dataset fully cover various situations, we collect the dataset in 7 different scenes, which contain 4 outdoor scenes and 3 indoor scenes.

After that, we manually label the location of the hand bounding box and label the fingertips and joints if the finger is visible. Additionally, we label not only visible fingertips but the relevant finger joints for we believe the joints can help the CNN to learn physical constraint of finger. We have released our dataset and we hope that our dataset can be helpful for the hand detection and fingertips detection and other hand related research in egocentric vision.

3.2. Dataset analysis

We compare the color distribution of an indoor scene and two outdoor scenes to demonstrate background and illumination conditions. We select one representative indoor scene and two representative outdoor scenes to calculate the RGB histogram. Figure 2(a) shows that the RGB histogram of an indoor scene is very different from that of an outdoor scene, Figure 2(b) shows that even in two outdoor scenes, the RGB histogram is obviously distinguishable from anothers. Since the RGB histogram is the description of background and illumination conditions, we can draw a conclusion that the background of the EgoGesture dataset is complex and various.



(a) Single hand gesture (b) Both hand gesture

Figure 3. Distribution of hand location.

According to human eye-tracking studies, gaze fixations are biased toward the center of natural scene stimuli (Center Bias [2, 27]) and it is a well-known fact that when humans take pictures they often frame the objects of interest near the center of the image, which called Center prior [3, 7]. To evaluate the Center Bias or Center Prior of the EgoGesture dataset, we analyze the distribution of hand location by calculating a 2D distribution matrix. Figure 3 is the 2D

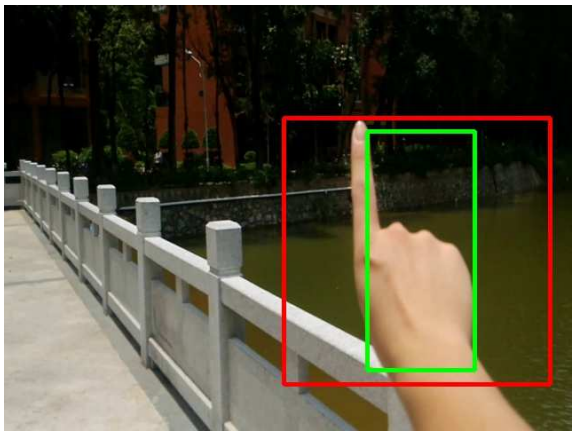


Figure 4. The hand detection result (green) and the input of YOLSE (red).

distribution matrix, which shows the distribution of hand location. We can see that no matter single hand gesture or bimanual gesture, the hand locates in the vision center, which proves that our dataset has the property named Center Bias or Center Prior.

Because of camera parameters, the hand shape variation or the distance between hand and camera, hands in video sequences are deformable. To analyze the scale of hands in vision, we calculate the ratio of the bounding box are to the frame size as the description of the hand scale. Among the single hand gesture, the scale of SingleFive is obviously bigger than the scale of SingleOne, which is cognitive while the scale of the bimanual gestures is about 25%.

It is shown that the EgoGesture dataset covers substantial data samples in various environments and dynamic hand shapes with the in-depth analysis on background, illumination conditions, hand location distribution and hand scale. We hope that our dataset can be beneficial for the hand detection and fingertips detection and other hand related research in egocentric vision.

4. YOLSE for multiple fingertip detection

The target of fingertip detection from single RGB images is to find the location of fingertips, which is similar to that of human pose estimation since the human pose estimation task is to find the location of body joints. [16] trained a HeatmapFusionNet to regress heatmaps of seven body joint positions instead of regressing the joint (x, y) positions directly. However, the number of body joints is fixed while the number of visible fingertips is different in different egocentric gestures. In addition, although directly regressing the coordinate of fingertip can work on the single gesture [8, 12], it turns out to be difficult to work in multi-gesture fingertip detection. The major reason is that different gestures have different number of visible fingertips but the output number is fixed in the previous designed network. In other words, they have to train more than one model if they want to detect fingertips from different gestures. What's worse, given an image without a hand, the network still output the coordinate in spite of the fact that there is no fingertip in the input image.

Since the above reasons, we proposed a heatmap-based FCN (Fully Convolution Network) named YOLSE (You Only Look what You Should See) to detect the multiple various number of fingertips from a RGB image and creatively proposed the fingermap to represent the most likely location and the appearance of the fingertip. Given the detected hand area, instead of directly regressing the fingertip coordinates, our network estimates the likelihood of each fingertip for each pixel and represents it as five individual channels of the output image (the last layer of network), called fingermap. Getting the output fingermaps, we find the location of maximum pixel value as the location of fingertip since the location of the maximum pixel value is where the fingertip most likely to locate in. We next discuss the YOLSE in detail.

4.1. YOLSE Input

The two-stage pipeline fingertip detection algorithms detect the fingertip by firstly detecting the hand area and take the area as the input of the fingertip detector in the second stage. Without appropriate process, the hand detection accuracy may have direct and important impact on the fingertip detection.

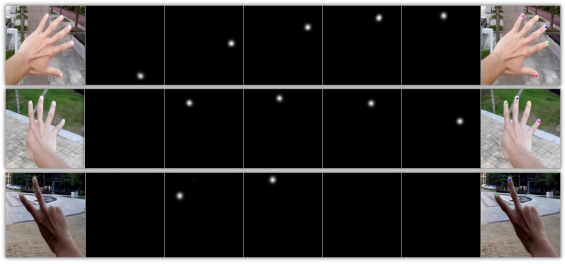


Figure 5. The fingermaps of gesture SingleTwo, SingleFour and SingleFive.

In this paper, we focus on the fingertip detection, to reduce the dependence on the hand detector. The input images of YOLSE are pre-processed as a 300 * 300 fix-sized square (bigger than the largest bounding box) centered on the hand bounding box (groundtruth or detected one). Subsequently, the area is cropped and resized to a 128 * 128 fixed size image as the input of our YOLSE network instead of directly cropping the bounding box as input. As Figure 4 shows, the green bounding box represents the bad hand detection result and the area within the 300 * 300 red square is the input area of the YOLSE. In this way, we can reduce the bad influence

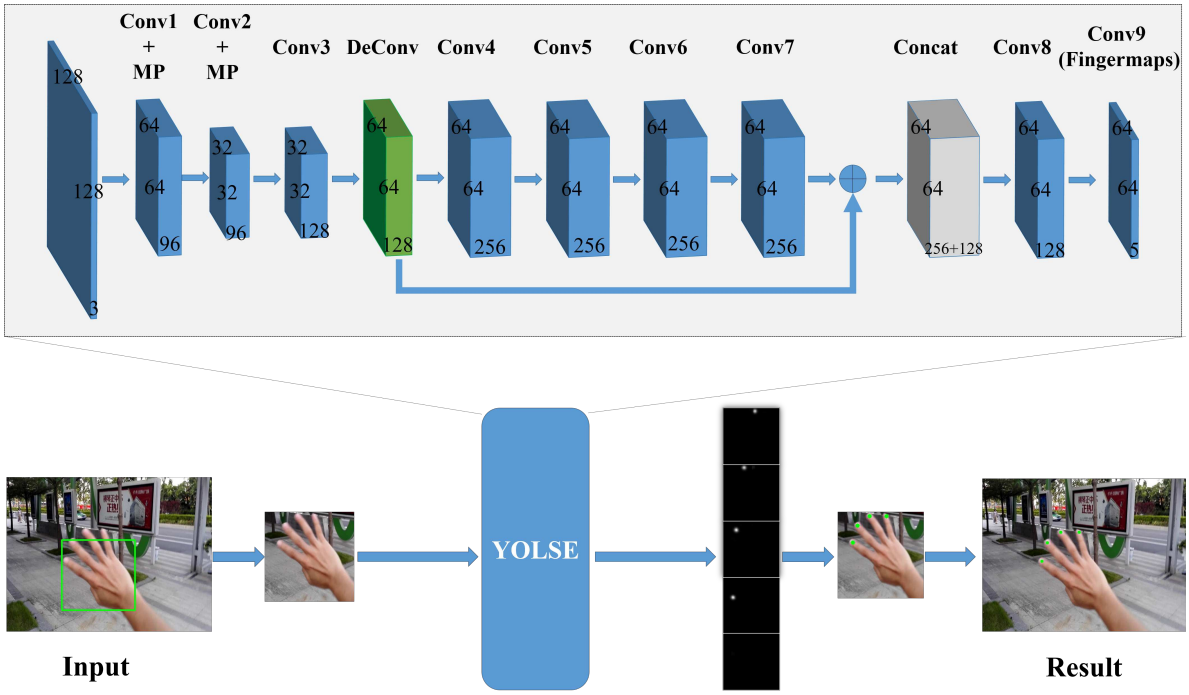


Figure 6. The architecture of YOLSE net and our fingertip detection framework.

on the fingertip detection that comes from bad hand detection result.

4.2. YOLSE fingermap

Inspired by human pose estimation, we use 2D probabilistic distribution to represent the possibility of fingertip location. Each fingertip has one corresponding detection estimation map, referred as fingermap. There is big difference here. In human pose estimation, the number of heatmaps is fixed, which is equal to the number of body joints. In our fingertip detection task, the number of visible fingertips is different in different gestures. To apply heatmap regression approach in fingertip detection on multi-gesture, we improve the heatmap to fit in fingertip detection by the following approach we propose so that we do not need to train multiple models for multiple gestures. In this paper, the output of the network is designed to be an image with five channels, which also called fingermaps. From the first to the fifth channel, they separately represent the fingermap of fingertip of thumb, index, middle, ring and pinkie. Except for placing a 2-dimension Gaussian with fixed variance (We set the variance to 1.5 in our experiment.) at the fingertip position, we set 0 as every pixel value of a fingermap if the relevant finger is invisible. For example, the thumb of the gesture SingleFour is invisible so that the fingermap of thumb is visually black for every pixel value is 0, which means the probability of every pixel in this fingermap to be thumb is 0. Figure 5 shows the fingermaps of gesture SingleTwo, SingleFour and SingleFive. In this way, the network can concentrate on the fingermap whose relevant fingertip is visible during training like human eyes do (thats why we called the network YOLSE). Getting an output fingermap, we find the location of the maximum pixel value and see whether the maximum is larger than a threshold P, which represents the minimal probability of being a fingertip. If the maximal pixel value is larger than P, we regard the position as the location of the fingertip. On the contrary, there is no fingertip detected if all pixel values are smaller than P.

4.3. YOLSE network

The HeatmapFusionNet [16] can be trained to detect fingertips but its computational load is fairly high and the detection error is comparable large. For the interaction purpose, the network speed has direct and important impact. Due to the deep network architecture and the large amount of big kernels, HeatmapFusionNet cannot meet the real-time requirement for real-time fingertip detection.

In consideration of both real-time performance and detection performance, we design a fully convolution network with less and smaller kernels to regress fingermaps of fingertip, as shown in Figure 6. To reduce the computational cost, we firstly set 128 * 128 as the input size of the proposed YOLSE and then we decrease the number of kernels

Method	Gesture				
	SinleOne	SingleTwo	SingleThree	SingleFour	SingleFive
GT-F [8]	7.12	7.08	7.08	7.43	7.62
FRCNN-F [8]	9.58	8.19	8.03	8.97	8.47
FRCNN-F End2End	6.53	7.11	7.60	6.90	7.78
GT-YOLSE	**3.68**	**2.90**	2.94	3.48	3.25
FRCNN-YOLSE	4.13	3.17	**2.75**	**3.33**	**3.21**

Table 1. The fingertip error (*px*) of YOLSE and direct regression approaches.

and reduce the kernel size. Since we found that 32 * 32 worked worse and the network was difficult to converge when set the size as 128 * 128, we use a deconvolution layer to keep the output size of last convolution layer is 64 * 64. In addition, because less and smaller kernels may lead to the decrease of fingertip detection performance, we concatenate the deconvolution layer and Conv7 layer to combine the low level features and high level features so that the network can ensure the nice performance in detecting the multiple various number of fingertips from a RGB image. Figure 6 shows the architecture of YOLSE net as well as the framework of our work. The output size of last convolution layer is 64 * 64 * 5, which represents the 5 fingermaps of an input image.

5. Experiments

5.1. Data augmentation

To reduce the risk of overfitting, data augmentation approaches are applied in our experiments. For the reason that we collect the single hand gestures with only right hand, we firstly mirror the image to generate left-hand samples for single hand gestures and mirror samples for bimanual gesture. Then we randomly sample one tenth of each gesture as the test set (We will release the image list for training the YOLSE). During training, we randomly change the brightness of images to simulate illumination changes and then randomly crop and rotate the image with a random angle within a specific range.

5.2. Fingertip detection

Considering fingertips as the objects to be detected, we have implemented the related detection algorithms for our fingertip problem, such as SSD [11] and Faster-RCNN (FRCNN) [19]. These algorithms all failed in detecting fingertips because fingertips are too small with variable numbers. Hence we propose YOLSE with high detection rate as well as weakening the dependence of fingertip detector on the hand detector.

To evaluate the performance of the proposed YOLSE net, we select five gestures (SingleOne to SingleFive) as the training data. We believe these five gestures cover enough various situations of fingertip number to demonstrate the algorithm capability. What we focus on is fingertip detection in this paper, so we assume the hand location is initially detected by any hand detection algorithms.During training, we optimize the network using Adam algorithm and we respectively set the batch size, learning rate, weight decay and momentum to 32, 0.1e-4, 0.0005 and 0.95.

Compare with direct regression approach. To compare with the algorithms that regress the coordinates of fingertip directly, we choose three different algorithms: GT-F, FRCNN-F and FRCNN-F End2End. The GT-F and FRCNN-F are two-stage frameworks, which firstly locate the hand area; the FRCNN-F End2End is an end-to-end framework to detect the hand and regress the coordinate of fingertip at the same time. However, all these algorithms locate the fingertips by directly regressing the coordinates of fingertips, they have to train five times for the five gestures since different gestures have different number of visible fingertips but the output number is fixed in the previous designed network. To have a fair comparison, we separately train the YOLSE with five gestures and compare the fingertip errorthe average euclidean distance between the predicted fingertips and the ground truth fingertips using two different hand detection results (manually labeled ground truth GT and FRCNN). As shown in Table 1, our YOLSE achieves the best performance comparing with some state-of-art fingertips detection algorithm.

According to [8], the result of fingertip detection relies on the hand detection accuracy because they crop the hand area by the detected hand bounding box and take it as the second stage input of their framework. The fact that the performance of FRCNN-F is worse than the GT-F in Table 1 also verifies the conclusion. To weaken the dependence of fingertip detector on the hand detector, in this paper, the source images are pre-processed with a 300 * 300 fix-sized square (bigger than bounding box) centered on the hand bounding box (GT or detected). Then we crop the hand area with the square and resize it to a 128 * 128 fixed size image as the input of the proposed YOLSE instead of

Threshold (*P*)	Method	Precision (%)	Recall (%)	Error (*px*)	Forward Time (*ms*)
0.2	**GT-HeatmapFushionNet [16]**	**99.09**	**99.26**	4.02	42.05
	GT-YOLSE	97.98	98.16	**3.94**	**15.15**
0.5	**GT-HeatmapFushionNet [16]**	**99.74**	**98.37**	3.91	42.05
	GT-YOLSE	99.32	96.73	**3.69**	**15.15**

Table 2. The comparison of YOLSE and heatmap-based approach.

directly cropping the hand area with the bounding box. The result (Table 1) not only shows that our heatmap-based fingertip detection framework achieves the best performance, but also testifies that our fingertip detector does not rely on the hand detection result since the performance of GT-YOLSE and FRCNN-YOLSE is comparable.

Compare with heatmap-based approach. We use the groundtruth bounding box to crop the hand area by applying the approach shown in Section 4.1. We trained our YOLSE net to compare the performance with the HeatmapFushionNet [16] and the comparison is shown in Table 2. The result shows that the fingertip detection performance of YOLSE is better than HeatmapFushionNet and achieve a promising fingertip detection error at about 3.69 pixels when the threshold *P* is 0.5. The precision and recall (When caculating the precision and recall, we consider the fingertip error which is larger than 15 pixels as wrong detection result.) decline a little because of less kernel number, smaller kernels size and the simpler network architecture. In addition, we compare the real-time performance between HeatmapFushionNet and YOLSE by calculating the average forward time on *Nvidia GTX980 Ti* with 6 GB memory. The average forward time of HeatmapFushionNet is about 42.05 *ms* while that of our YOLSE is about 15.15 *ms*. It shows that the YOLSE can meet the real-time requirement in the two-stage pipeline fingertip detection framework more.

6. Conclusion

In this paper, we focus on detecting variable number of visible fingertips in the egocentric video for further interaction. Firstly, we collect a large-scale dataset named EgoGesture, which contains 16 different hand gestures with manually labeled hand, fingertip and some key joints. The analysis of the dataset shows that our dataset is of diversity and representative for the fingertip and hand gesture related research in the egocentric vision. Then we design a novel fingermap-based FCN structure called YOLSE (You Only Look what You Should See) for fingertip detection from single RGB images. Given the detected hand region, our approach achieves the best performance comparing with some state-of-the-art fingertip detection algorithms. In our proposed frame, the error of hand detection has limited influence on the final fingertip detection result. According to our experiments, the fingertip detection error is 3.68 pixels in the 640 * 480 image and the average forward time is about 15.15 *ms*. In the future, we will include more gestures in our data set and develop some interaction applications.

7. Acknowledgement

This research is supported in part by the MSRA Research Collaboration Funds (FY16-RES-THEME-075), Fundamental Research Funds for Central Universities of China (2017MS050), GDSTP (2016A010101014, 2015B010101004, 2015B010130003, 2015B010131004, 2014A020208112, 2017A010101027), NSFC (61472144, 61673182), National Key Research & Development Plan of China (2016YFB1001405), GZSTP (201607010227, 201707010160).

References

[1] Georgia tech egocentric vision repository. `http://cbi.gatech.edu/egocentric/contact.htm`.

[2] M. Bindemann. Scene and screen center bias early eye movements in scene viewing. *Vision research*, 50(23):2577–2587, 2010.

[3] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):569–582, 2015.

[4] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang. Multi-context attention for human pose estimation. *arXiv preprint arXiv:1702.07432*, 2017.

[5] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. *arXiv preprint arXiv:1704.02463*, 2017.

[6] L. Ge, H. Liang, J. Yuan, and D. Thalmann. Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3593–3601, 2016.

[7] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):1915–1926, 2012.

[8] Y. Huang, X. Liu, X. Zhang, and L. Jin. A pointing gesture based egocentric interaction system: Dataset, approach and application. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 16–23, 2016.

[9] S. K. Kang, M. Y. Nam, and P. K. Rhee. Color based hand and finger detection technology for user interaction. In *Convergence and Hybrid Information Technology, 2008. ICHIT'08. International Conference on*, pages 229–236. IEEE, 2008.

[10] C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun. Real time hand pose estimation using depth sensors. In *Consumer depth cameras for computer vision*, pages 119–137. Springer, 2013.

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[12] X. Liu, Y. Huang, X. Zhang, and L. Jin. Fingertip in the eye: A cascaded cnn pipeline for the real-time fingertip detection in egocentric videos. *arXiv preprint arXiv:1511.02282*, 2015.

[13] M. Madadi, S. Escalera, X. Baro, and J. Gonzalez. End-to-end global to local cnn learning for hand pose recovery in depth data. *arXiv preprint arXiv:1705.09606*, 2017.

[14] M. Oberweger, P. Wohlhart, and V. Lepetit. Hands deep in deep learning for hand pose estimation. *arXiv preprint arXiv:1502.06807*, 2015.

[15] M. Oberweger, P. Wohlhart, and V. Lepetit. Training a feedback loop for hand pose estimation. In *IEEE International Conference on Computer Vision*, pages 3316–3324, 2016.

[16] T. Pfister, J. Charles, and A. Zisserman. Flowing convnets for human pose estimation in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1913–1921, 2015.

[17] J. L. Raheja, K. Das, and A. Chaudhary. Fingertip detection: a fast method with natural hand. *arXiv preprint arXiv:1212.0134*, 2012.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[19] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[20] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei, et al. Accurate, robust, and flexible real-time hand tracking. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3633–3642. ACM, 2015.

[21] A. Sinha, C. Choi, and K. Ramani. Deephand: Robust hand pose estimation by completing a matrix imputed with deep features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4150–4158, 2016.

[22] X. Suau, M. Alcoverro, A. López-Méndez, J. Ruiz-Hidalgo, and J. R. Casas. Real-time fingertip localization conditioned on hand gesture classification. *Image and Vision Computing*, 32(8):522–532, 2014.

[23] J. S. Supancic, G. Rogez, Y. Yang, J. Shotton, and D. Ramanan. Depth-based hand pose estimation: data, methods, and challenges. In *Proceedings of the IEEE international conference on computer vision*, pages 1868–1876, 2015.

[24] D. Tang, T.-H. Yu, and T.-K. Kim. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *Proceedings of the IEEE international conference on computer vision*, pages 3224–3231, 2013.

[25] J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (ToG)*, 33(5):169, 2014.

[26] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1653–1660, 2014.

[27] P.-H. Tseng, R. Carmi, I. G. Cameron, D. P. Munoz, and L. Itti. Quantifying center bias of observers in free viewing of dynamic natural scenes. *Journal of vision*, 9(7):4–4, 2009.

[28] A. Wetzler, R. Slossberg, and R. Kimmel. Rule of thumb: Deep derotation for improved fingertip detection. *arXiv preprint arXiv:1507.05726*, 2015.