Appears in: Intl. Conf. on Image Analysis and Processing (ICIAP), Trento, Sep 2019.

Hand Detection using Zoomed Neural Networks

Sergio R. Cruz and Antoni B. Chan

Department of Computer Science, City University of Hong Kong
{scruzgome2-c@my.cityu.edu.hk, abchan@cityu.edu.hk}

Abstract. The object detection problem has been widely focused due to the development of personal cameras allowing the general population to have access to high end cameras. This has resulted in cameras with various perspectives, one of which is the Egocentric Perspective, like the GoPro cameras. This new perspective opens the possibility of having hand detection as a special problem, due to the hands containing enough information to be detected and even for hand recognition and users's activity recognition. However, due to the perspective being new the databases are scarce, and most of them focus on generic object detection rather than hand detection. In this paper we address hand detection and hand disambiguation which focuses on detecting left and right hands as different objects.
This paper addresses these challenges by using the information of a left hand being the mirror image of the right hand for the hand disambiguation, and we also train a Neural Network to focus on the hand over all the image and another Neural Network to focus on the bottom area of the image, increasing the resolution as the hands go out of image, which is a characteristic of the hands in the Egocentric Perspective. In addition, we propose three Neural Network architectures using the hand and increase resolution bottom image information, and we compare them with current object/hand detection approaches.

Keywords: Neural Network · Hand detection · Hand disambiguation.

1 Introduction

As technologies in cameras advance, they become more accessible and they present new technological features. One of the new features is wearable cameras, where they can be placed on the person to record the person's perspective, or Egocentric Perspective. These cameras, like the GoPro cameras, have the ability to record videos with high resolution and high frame rate, which is needed as the person can move rapidly. This allows people to record activities that have fast movements like outdoor sports, e.g., hiking, surfing and biking in various scenarios, and daily activities, e.g., handling objects at home or washing the dishes.

This creates videos with swift movements and blurry nature. The first approaches on the egocentric perspective were in controlled environments by staying indoors walking through different rooms [5], or outdoors staying in the same place [1], but they neglect doing daily activities.

Due to the nature of this perspective, the camera wearer's hands appear in the image in reasonable size for information retrieval, and are the most consistent objects in the view since the person interacts with the world using their hands. However, the hand's characteristics make hand detection more challenging than generic objects as the hand's shape can change dramatically because the fingers and wrist have high degrees of freedom. Hand detection can be a first step for more complex analysis like activity recognition [17, 5], or hand pose estimation [25, 15], making it an essential step in the computer vision pipeline.

The first works to address the challenges in the egocentric perspective have used hand properties rather than the hands themselves to simplify the problem [10, 20]; however they do not detect the hand. For hand detection in the egocentric perspective, previous works have used segmentation [2, 11] by using a combination of low level features like color to detect skin-like features. However, such approaches also detect any skin-like regions, such as the arms.

In this work we approach hand detection from the egocentric perspective as an object disambiguation problem, by having the left and right hand as different objects, rather than treating them as a single type of object. We use Neural Networks for feature extraction and classification to detect the hands, as they have shown to have good performance for obtaining object information from images, and they are robust to illumination changes and blurriness, which is needed in this perspective.

We notice that current methods can detect other people's hands well, but often fails on the wearer's hands. This is because the wearer's hands often change shape drastically and are partially occluded (due to grasping of objects), or are only partially visible (due to the hands' proximity to the bottom of the camera's field of view).

Hence we devise an architecture with two parts: we train a Neural Network that focuses on the whole image for generic object detection, and another Neural Network that focuses on the bottom area of the image, where the wearer's hands have different properties (often appearing smaller due to exiting the image). We then fuse the results of both Neural Networks at different levels to show their behavior and performance. While it is possible to augment the training dataset by flipping left hands to look like right hands, here we do not flip them during training so as to allow context information to help improve the detection of both hands. Finally, we provide an ablation analysis to see how each of the changes contribute to the final performance.

2 Related Work

Due to cameras being more available to the public more data has been gathered using the egocentric perspective, resulting in various research works. Activity recognition [17, 5, 3, 16, 21–23] recognizes the activity the wearer is doing by using the generic objects found in the image. However, because the wearer's hands often handle the those objects, they sometimes move rapidly and appear blurry.

After object detection, object recognition and tracking [10, 8, 4, 20, 7] focus on using the detections to recognize the different appearances the object can take. This requires a more fine-grained feature usage which makes it more challenging.

Following this, research has been done on Summarization and Video Retrieval [14, 9, 13, 24], where the most representative parts of video sequences are extracted. This is usually achieved by focusing on the objects that contain the most information from the videos over time, in order to create a series of stories describing the videos sequences.

Combining hand detection and the egocentric perspective has been a recent problem, and only a few approaches have been developed. Early developments use hand characteristics like optical flow patterns to segment the image [20]. This takes advantage of the fact of the hands move noticeable differently than the background, as the hands follow the camera movement. However, this also makes any object that moves like a hand to be detected as well. In this pape, however we approach the problem using the shape to detect the hands.

Other approaches focus on color features to identify the skin using various approaches like scene-level feature probes [11] and random forests [10]. They provide a database that focuses on illumination change with drastic changes in the background by going indoors and outdoors, and changing rooms such as going into the kitchen and grabbing various objects. They approach the hand detection problem using segmentation, and select the best features that discriminate the hand pixels from the background. However this also identifies other pixels that resemble skin, such as the arms.

More recent work focuses on distinguishing the shape of the hands using Convolutional Neural Networks (CNN), which have had high performance on generic object detection. Bambach *et al.* [1] uses CNNs to recognize hands in various egocentric videos while multiple people interact with each other. They generate bounding box proposals using the probability of the location, bounding box size and color-like features, and apply a CNN to score them. Then they combine the obtained bounding boxes and the segmentation of the image using previous works to provide a more refined segmentation.

They focus on the hands of two people interacting as they play board games. They collect a database with various pairs of people interacting in indoor/outdoor places, and add illumination changes and background variability. Our approach addresses improving the detection of the wearer's hands, which can change shape drastically and be partially occluded (due to grasping of objects), or only partially visible (due to the proximity to the edge of the camera's field of view).

Finally, much research has focused on generic object detection based on CNNs using an end-to-end approach [18, 12, 19], where the CNN extracts the features, proposes object regions, and classifies the regions within the same network. In this paper we extend these approaches to egocentric hand detection.

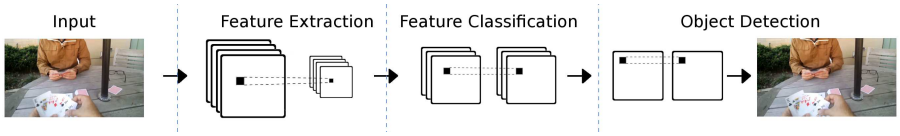

Fig. 1. Sections of the YOLO [18] neural network.

3 YOLO detector

Due to the many variations of the hand shape caused by the egocentric perspective, hand detection and disambiguation is still a challenging problem even for approaches based on CNNs. The wearer's hands in this perspective move in and out of the image, sometimes only occupying a small area or even completely disappearing, which causes problems for standard CNN approaches. To address this issue, we devise an architecture using a backbone object detector that can specifically detect the wearer's hands.

We use YOLOv2 [18] as the backbone object detector due to its high performance and speed (we denote it as simply YOLO here). For YOLO, there is a limit on how large the input image resolution can be, as higher resolutions would only lead to an increase of false positives and decrease of overall performance. The YOLO architecture contains 3 sections (see Figure 1): 1) feature extraction consisting of 18 convolutional layers and 5 max-pooling layers, resulting in a decrease in feature map resolution; 2) feature classification consisting of 8 convolutional layers; 3) object detection consisting of a final convolutional layer for bounding box regression and object classification.

The YOLO network has 5 outputs representing the bounding box and confidence, and the classification outputs. The bounding box outputs are the x coordinate (t_x), y coordinate (t_y), width (t_w) and height (t_h). Since this paper focuses on hand disambiguation we set the classification outputs as $C = 4$, as for hand disambiguation we detect left and right hands for wearer and other people as all different objects.

3.1 Detection using zoom information

YOLO has a fixed resolution which has a limit on the size of the objects it is able to detect. If the object is too small it cannot extract enough information, which is a characteristic of some hand regions in the egocentric perspective. We address this using a second YOLO that focuses on the bottom of the image where the hands appear the smallest due to partial occlusion. We name it as ZOOM to distinguish from YOLO. We construct ZOOM to be able to see the small objects a standard YOLO cannot see, as seen in Figure 2 where we highlight in green the area the ZOOM focuses on. We train ZOOM using only the ground truth bounding boxes that completely are contained in the bottom area of the image, creating a second smaller dataset with only small size hand regions.

Fig. 2. ZOOM takes as input one fourth of the image (green), and the bounding boxes that entirely fall into that area (red box) as ground truth.

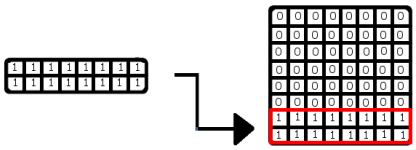

Fig. 3. Grid expansion from the ZOOM output grid cell by adding the upper grid cells and setting them to 0.

3.2 Fusion architectures

We propose to combine YOLO and ZOOM as follows. First, we train YOLO separately as the standard framework, with full images and all the ground truth bounding boxes. Second, we train ZOOM separately using the hand bounding boxes in the bottom area of the image. The two trained networks form a two-stream architecture: the 1st stream is the standard YOLO networks for detecting hands, and the 2nd stream is the ZOOM network.

We fuse the two streams by concatenating the features at a given level into one big feature map, and then fine-tuning the rest of the layers. Since the second stream represents the bottom quarter of the first stream, we need to expand it and fill in the missing grid cells before the concatenation. First we apply the CNNs to obtain the grid from YOLO and ZOOM, then we expand the ZOOM grid by four times and put the new cells to 0, as seen in Figure 3.

Specifically, we set the YOLO input resolution for the stream to be 384 so that the output grid size is 12×12. We then set the ZOOM input resolution so that the ZOOM output grid after the expansion is a multiple of the YOLO output grid in the width and height (up to twice the size of the YOLO grid). We set different widths to be of 384 and 768 pixels denoted as W384 and W768 respectively, and we set different heights to be of 96 and 192 pixels denoted as H96 and H192 respectively, resulting in 4 different resolutions for the ZOOM network. We found increasing the resolution even more decreases the performance.

We then concatenate the YOLO grid cells with the corresponding ZOOM grid cells of higher resolution, as seen in Figure 4. These YOLO and ZOOM resolutions ensure that grid cells from both networks can be matched without any spatial displacement,. as the ZOOM output grid sizes for W384 and W768 are 12 and 24 respectively, and the output grid sizes of H96 and H192 are 3 and 6 respectively.

To combine the two streams we propose three levels of fusion, as seen in Figure 5. The first method concatenates both streams after the Feature Extraction section, and then fine-tunes the rest of the convolutions on the feature combination on the Feature Classification and Object Detection sections, which we

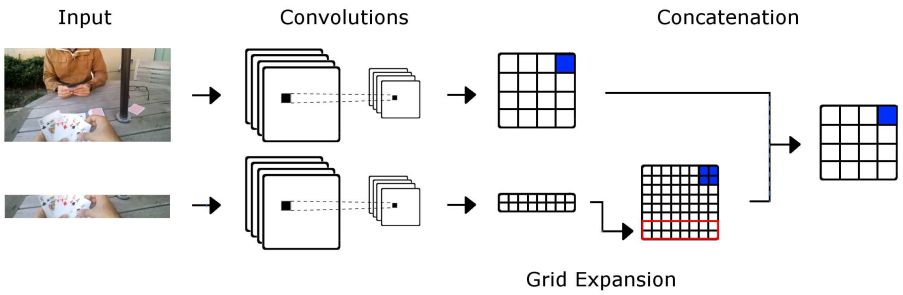

Fig. 4. Feature grid output from convolutions. ZOOM grid (red box) is expanded setting all the extra grid cells with 0 and then concatenates (blue squares) with the YOLO stream.

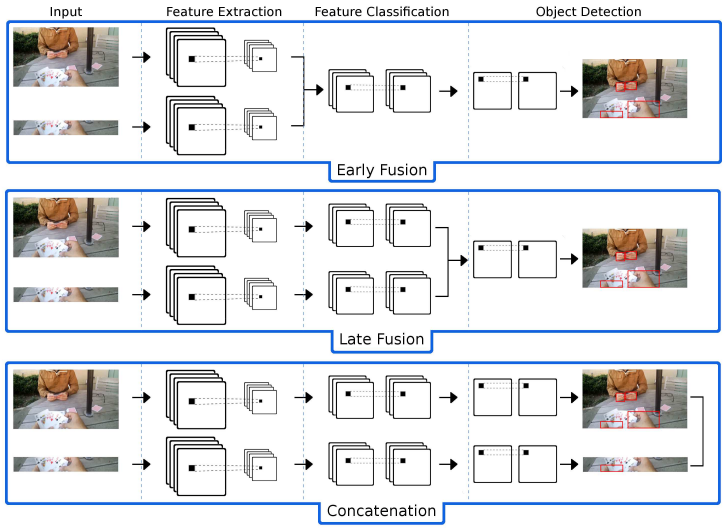

Fig. 5. Proposed network architectures combining the already trained YOLO and ZOOM streams.

denote as "Early Fusion". The second method concatenates both streams after the Feature Classification section, and fine-tunes the final convolution on the Object Detection section, which we denote as "Late Fusion". Finally, the last combination concatenates the detection bounding boxes from the YOLO and ZOOM networks, and we denote it as "Concat".

3.3 Detection using non flipped information (NoFlip)

A subsequent step after hand detection is hand disambiguation, where the left and right hands are classified as different objects, but due to the hand shape variability it makes it a challenging task. Standard object detection CNNs will horizontal flip images during training as data augmentation. However, this is counter productive for hand disambiguation, since the flipped left-hand will look like the right hand. Thus, we do not allow flipping of the the input images during training, which we denote as "NoFlip".

4 Hand Disambiguation Experiments

In this section we show a comparison between the proposed methods and current egocentric hand/object detection approaches, as well as an ablation study.

4.1 Experiment Setup

Our experiment uses the EgoHands dataset by Bambach *et al.* [1], which consists of pairs of people playing board games in different locations. The dataset contains 48 videos, with 4,800 annotated frames with pixel-level masks (15,053 annotated hands). The hand disambiguation task is a object detection task with four classes: left and right hand of the wearer, and left and right hand of the other person. We test our three proposed Neural Networks using zoomed information:

- **EarlyFusion** combines the YOLO detector stream with the ZOOM stream after the "Feature Extraction" stage.
- **LateFusion** combines the YOLO detector stream and ZOOM stream after the "Feature Classification stage".
- **Concat** is the concatenation of the hand detections from the YOLO and ZOOM streams.

For all three proposed Neural Networks we using the NoFlip configuration with different resolutions denoted as **W384** and **W768** for the width, and **H96** and **H192** for the height, e.g. **EarlyFusion W384 H192 NoFlip**. We compare against 4 baseline methods consisting of generic object/hand detection methods based on neural networks:

- **Bambach *et al.*** [1] generates object proposals using probability distributions of hand properties, and then uses a neural network to classify the proposals.
- **Faster R-CNN** [19] is a deep CNN that uses the region proposal network (RPN) to extract features from the image and then the Fast R-CNN Neural Network [6] on a fully convolutional network trained end-to-end. For the feature extraction CNN we use the VGG-16 version.
- **SSD** [12] uses a deep Neural Network for feature extraction and classification, and is trained end-to-end. It uses predefined bounding boxes with different aspect ratios and scales to predict the ground truth. We use the author's code and set the input resolution as 300×300 and keep the other settings as their defaults. During training we set the base learning rate to 0.0001.
- **YOLO** [18] uses a single Neural Network to predict bounding boxes and class probabilities, and is trained end-to-end. We use the code provided by the author with a 416×416 input resolution and keep the other settings as their defaults.

After running the detection algorithms, we use non-maximum suppression with overlap of 0.5 as post-processing.

Table 1. Experiment results for hand disambiguation. OL/OR are other's left/right hand, and WL/WR are wearer's left/right hand.

Method	Average Precision					Recall
	OL	OR	WL	WR	All	
Bambach *et al.* [12]	0.556	0.698	0.596	0.553	0.587	0.771
Faster [19]	0.754	0.809	0.681	0.582	0.745	0.838
SSD [12]	0.839	0.870	0.847	0.700	0.834	0.930
YOLO [18]	0.869	0.894	0.771	0.694	0.790	0.890
EarlyFusion H96 W384 NoFlip	0.905	0.906	0.899	0.810	0.902	0.929
LateFusion H96 W768 NoFlip	0.905	0.907	0.899	0.806	0.903	0.929
Concat H192 W384 NoFlip	0.906	0.907	0.905	0.896	0.904	0.939

4.2 Hand Disambiguation Results

Table 1 shows the breakdown of hand disambiguation performance using average precision (AP) and recall on all the hands. We first examine the baseline methods. Bambach *et al.* [1] has similar performance across all hands, showing their features are not affected by the number of instances and the hand sizes in the dataset. However the average precision is low while the recall is high, showing that the Neural Network classifier has problems telling apart hands.

Faster R-CNN has a considerable decrease in performance on the wearer's hands (WL/WR), with the lowest being the right hand. This demonstrates that the baseline object detectors cannot handle small objects well, as well as they may be affected by the low number of training samples.

SSD [12] has similar performance across hands except on the wearer's right hand (WR), and outperforms Faster R-CNN, showing that the fine-grained features can increase performance. Finally, YOLO has lower recall and overall average precision than SSD. However the average precision on the Other person's hands is higher, which shows that YOLO does well on bigger objects, but encounters difficulties with small objects, more specifically on the wearer's left hand (WL).

Compared to the previous methods, our proposed methods have higher average precision (AP) on all hands, while maintaining similar or better recall. The biggest increase in AP is with the wearer's hands (more than 0.1), which demonstrates the impact of our proposed fusion method. The main difference in the performance among our fusions method is on the wearer's hands, as the wearer's hands go in and out of the camera view more often than the Other's hands, which are mainly completely in the view.

The "Concat" fusion has the best performance with a noticeable AP increase on the wearer's right hand. In contrast, Late and Early fusion have lower PA on the wearer's right hand. This suggests the combination of ZOOM and YOLO can obtain better performance if they are learned separately and making them more specialized, rather than having layers on the Neural Network to learn.

Table 2. Ablation study on variants of our proposed method. The right block of results uses NoFlip setting, while the left-block uses image flipping during training. OL/OR are other's left/right hand, and WL/WR are wearer's left/right hand.

Method	Flip						NoFlip					
	Average Precision					Recall	Average Precision					Recall
	OL	OR	WL	WR	All		OL	OR	WL	WR	All	
YOLO W384	0.796	0.833	0.773	0.675	0.784	0.873	0.902	0.907	0.898	0.806	0.899	0.919
YOLO W416	0.869	0.894	0.771	0.694	0.790	0.890	0.901	0.904	0.899	0.812	0.900	0.927
EarlyFusion H96 W384	0.892	0.904	0.897	0.809	0.890	0.915	0.905	0.906	0.899	0.810	0.902	0.929
EarlyFusion H96 W768	0.898	0.905	0.884	0.803	0.888	0.912	0.905	0.906	0.899	0.810	0.902	0.928
EarlyFusion H192 W384	0.901	0.903	0.882	0.801	0.895	0.918	0.905	0.906	0.897	0.807	0.901	0.928
EarlyFusion H192 W768	0.900	0.902	0.880	0.744	0.893	0.916	0.905	0.906	0.897	0.809	0.901	0.926
LateFusion H96 W384	0.803	0.883	0.833	0.694	0.788	0.890	0.906	0.907	0.896	0.803	0.902	0.926
LateFusion H96 W768	0.795	0.882	0.827	0.693	0.793	0.884	0.905	0.907	0.899	0.806	0.903	0.929
LateFusion H192 W384	0.798	0.885	0.848	0.701	0.795	0.891	0.906	0.907	0.897	0.806	0.902	0.926
LateFusion H192 W768	0.792	0.883	0.837	0.698	0.792	0.890	0.906	0.907	0.896	0.802	0.902	0.928
Concat H96 W384	0.798	0.879	0.816	0.683	0.770	0.898	0.906	0.907	0.904	0.893	0.903	0.938
Concat H96 W768	0.798	0.879	0.796	0.680	0.766	0.899	0.906	0.907	0.904	0.889	0.903	0.937
Concat H192 W384	0.798	0.879	0.817	0.681	0.773	0.895	0.906	0.907	0.905	0.896	0.904	0.939
Concat H192 W768	0.798	0.879	0.793	0.684	0.769	0.900	0.906	0.907	0.905	0.887	0.904	0.938

5 Ablation Study

Table 2 presents the breakdown of the impact of our proposed fusions for various configurations. The YOLO W384 NoFlip configuration increases the AP of all hands, showing the main difference between left and right hand shapes is its flipped characteristics. However the lowest AP are found in the wearer's hands, especially on the right hand, which shows the difficulty YOLO still has on these instances. The increase of the regular YOLO W416 over W384 shows the increase resolution is able help on distinguishing more hand shapes, as the main increase occurs on the Other's hands.

The proposed fusion networks have increased recall and overall performance, compared to standard YOLO, which shows that the ZOOM network is able to find more of the Wearer's hands, as they focus on the hands which go in and out of the bottom of the image, . Among the different resolutions, using twice the resolution for the height (H192) the ZOOM Networks are able to find the most hands, showing the main struggle of the feature extraction as the hands leave the image is the decrease of the hand's height.

Using the Flip configuration for training, the performance increases with earlier fusion of the two streams. This suggests that solving the hand disambiguation task using the Flip configuration requires complex patterns extracted between the two streams.

This Flip configuration is also affected by the small number of instances on the training dataset, since there are not enough instances to learn such complex patterns.

Using the NoFlip configuration yields the opposite behavior compared to the Flip configuration – better performance is achieved with later fusion of the streams. This implies that the two streams can work well independently, and detection of Other's hands and wearer's hands do not affect each other significantly.

This shows the main challenge of the hands on the egocentric perspective is the different shapes between the left and right hands (they are flipped versions of each other), which is addressed by the NoFlip configuration, showing the highest performance compared to the Flip configuration of the different fusions. The addition of the fusions to the NoFlip configuration is able to find the most challenging hands, the Wearer's hands as they leave the image, especially the right hand.

6 Conclusions

We have proposed a variation of the YOLO Neural Network focusing on a zoomed area of the image, denoted as ZOOM, using the unique characteristics found on the hands in the egocentric perspective. We have proposed three different joint Neural Network architectures combining YOLO and ZOOM to improve on the disambiguation performance and we present a comparison with object/hand baseline detection methods. We showed how the extracted zoomed information and training without the flipped setting on the Neural Network can help finding small hand regions and disambiguating them.

References

1. Bambach, S., Lee, S., Crandall, D., Yu, C.: Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In: ICCV (2015)
2. Betancourt, A.: A sequential classifier for hand detection in the framework of egocentric vision. In: CVPRW '14. pp. 600–605 (2014)
3. Fathi, A., Hodgins, J.K., Rehg, J.M.: Social interactions: A first-person perspective. In: CVPR. pp. 1226–1233 (2012)
4. Fathi, A., Ren, X., Rehg, J.M.: Learning to recognize objects in egocentric activities. pp. 3281–3288. CVPR '11 (2011)
5. Fathi, A., Farhadi, A.: Understanding egocentric activities. pp. 407–414. ICCV '11 (2011)
6. Girshick, R.: Fast r-cnn. In: International Conference on Computer Vision (ICCV) (2015)
7. Kolsch, M., Turk, M.: Robust hand detection. In: Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings. pp. 614–619 (2004)
8. Lee, S., Bambach, S., Crandall, D.J., Franchak, J.M., Yu, C.: This hand is my hand: A probabilistic approach to hand disambiguation in egocentric video. In: CVPRW. pp. 557–564 (2014)
9. Lee, Y.J., Ghosh, J., Grauman, K.: Discovering important people and objects for egocentric video summarization. In: CVPR. pp. 1346–1353 (2012)

10. Li, C., Kitani, K.M.: Pixel-level hand detection in ego-centric videos. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. pp. 3570–3577 (June 2013). https://doi.org/10.1109/CVPR.2013.458
11. Li, C., Kitani, K.M.: Model recommendation with virtual probes for egocentric hand detection. pp. 2624–2631. ICCV '13 (2013)
12. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: Single shot multibox detector. In: ECCV (2016)
13. Lu, Z., Grauman, K.: Story-driven summarization for egocentric video. pp. 2714–2721. CVPR '13 (2013)
14. Min, W., Li, X., Tan, C., Mandal, B., Li, L., Lim, J.H.: Efficient retrieval from large-scale egocentric visual data using a sparse graph representation. pp. 541–548. CVPRW '14 (2014)
15. Mueller, F., Bernard, F., Sotnychenko, O., Mehta, D., Sridhar, S., Casas, D., Theobalt, C.: Ganerated hands for real-time 3d hand tracking from monocular rgb. In: CVPR (2018)
16. Pirsiavash, H., Ramanan, D.: Detecting activities of daily living in first-person camera views. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE (2012)
17. Poleg, Y., Arora, C., Peleg, S.: Temporal segmentation of egocentric videos. In: CVPR (2014)
18. Redmon, J., Farhadi, A.: Yolo9000: Better, faster, stronger. arXiv preprint arXiv:1612.08242 (2016)
19. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. CoRR (2015)
20. Ren, X., Gu, C.: Figure-ground segmentation improves handled object recognition in egocentric video. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. pp. 3137–3144 (June 2010)
21. Ryoo, M.S., Matthies, L.: First-person activity recognition: What are they doing to me? In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on (2013)
22. Ryoo, M.S., Matthies, L.: First-person activity recognition: Feature, temporal structure, and prediction. International Journal of Computer Vision pp. 307–328 (2016)
23. Song, S., Chandrasekhar, V., Mandal, B., Li, L., Lim, J.H., Sateesh Babu, G., Phyo San, P., Cheung, N.M.: Multimodal multi-stream deep learning for egocentric activity recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (June 2016)
24. Spriggs, E.H., De la Torre Frade, F., Hebert , M.: Temporal segmentation and activity classification from first-person sensing. In: IEEE Workshop on Egocentric Vision, CVPR 2009 (June 2009)
25. Spurr, A., Song, J., Park, S., Hilliges, O.: Cross-modal deep variational hand pose estimation. In: CVPR (2018)