Cascaded Interactional Targeting Network for Egocentric Video Analysis

Yang Zhou[1], Bingbing Ni[2], Richang Hong[3], Xiaokang Yang[2], and Qi Tian[1]
[1]University of Texas at San Antonio, US
[2]Shanghai Jiao Tong University, China
[3]HeFei University of Technology, China
myh511@my.utsa.edu nibingbing@sjtu.edu.cn hongrc@hfut.edu.cn
xkyang@sjtu.edu.cn qi.tian@utsa.edu

Abstract

Knowing how hands move and what object is being manipulated are two key sub-tasks for analyzing first-person (egocentric) action. However, lack of fully annotated hand data as well as imprecise foreground segmentation make either sub-task challenging. This work aims to explicitly address these two issues via introducing a cascaded interactional targeting (i.e., infer both hand and active object regions) deep neural network. Firstly, a novel EM-like learning framework is proposed to train the pixel-level deep convolutional neural network (DCNN) by seamlessly integrating weakly supervised data (i.e., massive bounding box annotations) with a small set of strongly supervised data (i.e., fully annotated hand segmentation maps) to achieve state-of-the-art hand segmentation performance. Secondly, the resulting high-quality hand segmentation maps are further paired with the corresponding motion maps and object feature maps, in order to explore the contextual information among object, motion and hand to generate interactional foreground regions (operated objects). The resulting interactional target maps (hand + active object) from our cascaded DCNN are further utilized to form discriminative action representation. Experiments show that our framework has achieved the state-of-the-art egocentric action recognition performance on the benchmark dataset Activities of Daily Living (ADL).

1. Introduction

Recent years have witnessed the emergence of first-person (egocentric) action analysis due to its various applications in assisted daily living, medical surveillance and smart home [11, 38]. Daily living egocentric videos involve a large amount of manipulation actions. The key challenges are irrelevant objects-of-interest and the noisy background motions. Therefore, the key of addressing these issues is to successfully segment out hand region and active region (*i.e.*, the objects-of-interest region).

Prior art [34, 25] in egocentric action analysis have paid the most attention to these sub-tasks. Ren *et al*. [34] quantitatively analyzed the feasibilities and challenges of the egocentric recognition of handled objects. It was pointed out that hand and motion information are the keys to solve the egocentric video recognition problem. Pirsiavash *et al*. [31] used a temporal pyramid for both passive and active objects as the action representation, and they suggested that the daily living egocentric video understanding are "all about the objects being interacted with". McCandless *et al*. [25] applied an "object-centric" scheme to automatically select some representative spatio-temporal partitions from a pool of pre-detected partitions.

However, there are two major difficulties in hand and foreground object segmentation. Firstly, in egocentric videos, the segmentation map for a non-rigid object like hand is more helpful in later processing (*e.g.*, motion feature pooling) than object bounding box information. Unfortunately, previous methods for hand detection in egocentric video analysis mainly adopt the hand detection method (*i.e.*, bounding box detection) [26, 16, 15], which is not precise enough to model hand movements. This is mainly due to the lack of pixel-level hand annotations for training good hand segmentation models. In fact, it is not feasible to get a large scale per-pixel annotated hand dataset because it requires intensive human labor. Moreover, pixel-level hand detection/segmentation is challenging due to large illumination changes and hand deformations. Secondly, most of previous works separately model the hands, foreground objects and motion information. Due to the noisy background motion, highly frequent occlusion, and object deformation, jointly detecting/segmenting these objects is hard. We find the rich contextual information could be explored to enhance the detection/segmentation. Two important observations are that the hand information are helpful for localizing the hand-manipulated objects, and the motion information are useful for detecting the foreground objects. Motivated by these observations, we propose a hybrid/cascaded end-to-end deep

convolutional neural network to jointly infer the hand maps and manipulated foreground object maps.

On one hand, we propose a novel end-to-end trainable semantic parsing network for hand segmentation. In order to tackle the problem of insufficient fully annotated (*i.e*., per-pixel annotated) hand maps, we develop an EM-like training method to augment the semi-supervised data, *i.e*., a small set of fully annotated hand segmentation data and a large number of hand bounding box data. In particular, in the E-step, we firstly generate a number of hand map proposals by the traditional hand segmentation method such as [5, 14, 13]. Then we use our trained deep semantic parsing neural network [24, 4] to select the best hand candidate, (*i.e*., the hand candidate has the largest overlap with the predicted hand segmentation map). In the M-step, the selected hand candidates are considered as new ground truth which are utilized to further fine-tune the hand segmentation neural network. The converged network model parameters are used as the ultimate hand segmentation model. We evaluate our approach on the Georgia Tech Egocentric Activity (GTEA) [9] dataset. In the experiments, we show that our proposed pixel-level hand detection could handle some difficult cases such as large illumination changes, or hand deformations in egocentric videos.

On the other hand, we propose a second end-to-end deep convolutional network to maximally utilize the contextual information among hand, foreground object, and motion for interactional foreground object detection. More specifically, the network inputs are the strongest object feature maps from the convolutional layer of AlexNet [18], the hand segmentation maps detected by our proposed end-to-end semantic parsing network, and the optical flow motion maps. The convolutional filters connected to each pair of the three types of maps are learned to explore the contextual information and generate the interactional foreground object maps. Finally, based on the detected foreground object maps, we pack both object-centric features and the locally pooled motion features into one unified action representation, which is used for the action classification task. The action recognition framework is extensively evaluated on the egocentric video benchmark dataset Activities of Daily Living (ADL) [31]. We show that our proposed framework significantly outperforms the state-of-the-art algorithms in terms of action classification performance.

We summarize our contributions as two-fold: 1) we propose a novel iterative training scheme for a pixel-to-pixel hand segmentation DCNN. Our work *transforms* the weakly supervised hand bounding boxes into the strongly supervised hand segmentations, which saves a large amount of human labor for per-pixel annotation; 2) contextual information among the corresponding hand segmentation map, object feature map, and motion map are jointly explored in a DCNN architecture to generate accurate interactional foreground regions, *i.e*., the area of manipulated objects.

2. Related Work

2.1. Pixel-level Hand Detection

The approaches to generate pixel-level detections of hand regions mainly fall into the following three categories: (1) the appearance based methods [40, 2, 17, 20, 13] use the appearance features such as color or texture to detect static and dynamic appearance of skin; (2) the global appearance-based models [33, 19, 42, 29] detect hands by matching against the global hand templates under different configurations; (3) the motion-based approaches [39, 12, 9] explicitly take into account the ego-motion of the camera by assuming that hands (foreground) and the background have different motion or appearance statistics. The focus of our work is different, *i.e*., to design an effective training scheme that can utilize weakly supervised hand bounding box data (easily obtainable) with a small set of strongly supervised hand segmentation map data (expensive) to facilitate the pixel-to-pixel hand segmentation DCNN training.

2.2. Egocentric Action Recognition

Currently, there are mainly four types of methods focusing on the egocentric action recognition. First of all, the objects manipulated by human hands in egocentric videos are modeled [6, 31, 25]. Secondly, it is suggested that gaze location is an important cue for egocentric activity recognition [8, 21], but this fine-grained information is difficult for detection. Thirdly, human-human interaction or human-object-human interaction [7, 36] is important for egocentric action recognition. For example, Ryoo *et al*. [36] integrated global and local motion information to model interaction-level human activities. Finally, motion features also play an important role in egocentric action analysis [27, 32, 37], which is consistent with the general third-person action recognition scenario [23, 43, 44, 45, 30]. Ying *et al*. [22] combines multiple cues including object, motion, head movement, hand and gaze information to achieve the state-of-the-art, which is a sound work for egocentric information fusion. However, these information are not always feasible from pure wearable camera. We focus on the problem of recognizing single-human activities of daily living, where there is a large amount of manipulation actions. In such case, neither gaze information nor human-human interaction modeling could well solve the problem. We argue that it is necessary to detect foreground interactional (manipulated) objects, and combine both object cue and motion cue for action representation.

3. Cascaded Interactional Targeting Network

Two key difficulties that prevent the egocentric action recognition from higher accuracy are the deficiency in segmenting/identifying hand and interactional foreground object. Therefore, the key idea of this work is two cascaded end-to-end DCNNs which identify both hand regions

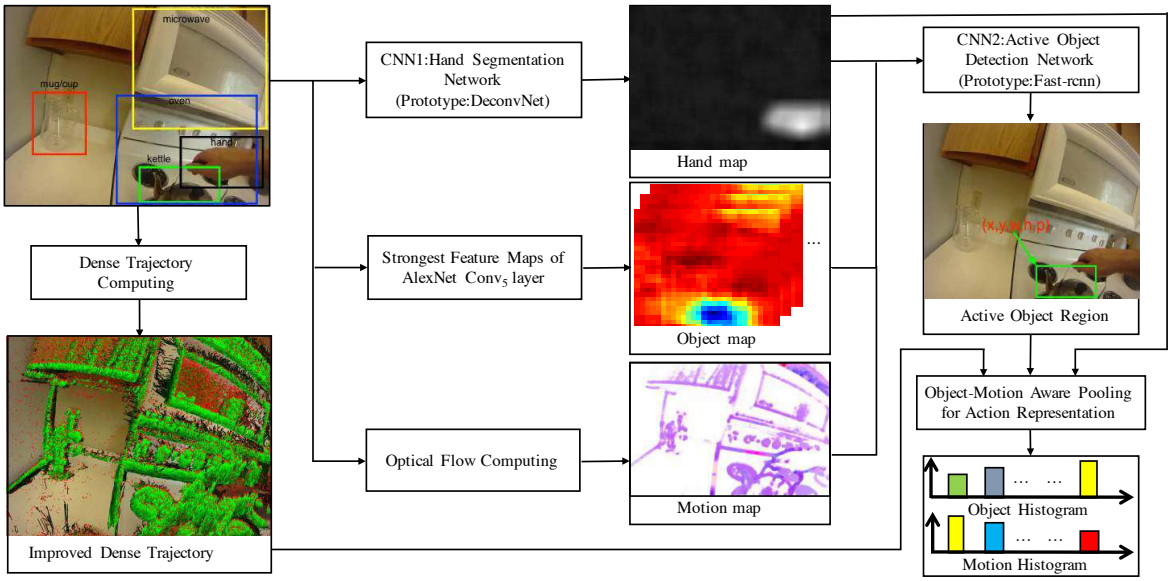


Figure 1: The cascaded neural network to collaboratively infer the hand segmentation maps and manipulated foreground objects. It includes the hand segmentation network CNN1 and the active object detection network CNN2.

and foreground (active) object regions to facilitate egocentric action analysis. The cascaded DCNN infrastructure is shown in Figure 1, where the resulting hand segmentation maps from the first DCNN (called pixel-to-pixel hand segmentation sub-network, *i.e.*, CNN1 in Figure 1) are input to the second DCNN (called active/interactional object detection sub-network, *i.e.*, CNN2 in Figure 1) to infer the interactional foreground region. Outputs of both sub-networks are further utilized to form the active object histogram and locally pooled motion histogram (on interactional foreground regions) for action representation.

3.1. Pixel-level Hand Segmentation

Detecting and segmenting hand regions up to pixel-level precision is challenging [39, 13, 9, 20]. This is partly due to the fact that ground-truth images with pixel-level hand segmentations are rare. However, bounding box based hand annotations are easily obtainable, and rich in public available datasets such as GTEA [9]. Inspired by these observations, we propose an expectation-maximization style algorithm to train a pixel-to-pixel hand segmentation network by fully utilizing the weakly supervised data (*i.e.*, hand bounding boxes). This algorithm **starts** from a small set of fully annotated hand segmentation maps, and iteratively **selects** and **adds** *good hypothesized* hand maps (augmented from the weakly supervised hand data) to gradually **refine** the end-to-end hand segmentation network, in an expectation-maximization manner.

Network Architecture. To generate hand segmentation maps based on image input, we adopt the DeconvNet deep network [28] for semantic parsing. This model improves the prior FCN model [24] (*i.e.*, using coarse bilinear interpolation as deconvolution procedure) with more sophisticated deconvolution and un-pooling layers. Specifically, we employ the VGG-16 net [41] as our baseline convolutional network, and we initialize this network with 1500 fully annotated hand images (pixel-level annotations) from the GTEA dataset [9]. The network parameter set is denoted by θ. The network is illustrated in Figure 2. This network contains two symmetric parts of the VGG-16 net, *i.e.*, local-to-global convolutional network and the mirroring global-to-local deconvolutional network. More details of the network can be seen in [28].

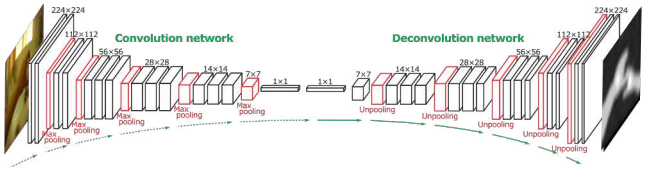


Figure 2: We use the end-to-end DeconvNet [28] as our baseline semantic parsing network for hand segmentation. The network outputs the hand probability score map, the brightness indicates the likelihood of hand region.

Our algorithm to learn the above network is as follows: firstly, we use the fully annotated hand image samples to initialize the DCNN, we fine-tune the weights in the convolutional layers using VGG-16 model pre-trained on ILSVRC dataset, while the weights in the de-convolutional network are initialized with zero-mean Gaussians. Then we employ an expectation-maximization style training procedure, to alternate the following two steps: 1) with the network parameters θ fixed, we seek the best hand map proposals for the weakly supervised data; and 2) we augment the fully annotated dataset with the newly identified hand maps from the weakly supervised data, and refine the model parameter set

θ. Each step is detailed as follows. Our iterative network training procedure is also elaborated in Algorithm 1.

E-Step: Seek the Best Hand Map Proposal. With the network parameters θ fixed, we generate a set of hand map/mask hypotheses from each weakly supervised hand image (*i.e.*, with bounding box), then search the best hypothesis/proposal for next iteration of hand segmentation network training. In particular, we firstly apply super-pixel segmentation using SLIC [1] onto each bounding box image I_n. We then represent each super-pixel with a concatenated feature vector consisting of HSV color histogram and Gabor filter texture histogram. A linear support vector machine classifier is trained and applied to calculate the hand detection scores for each super-pixel. We build our training object patch (super-pixel) dataset by randomly annotating 50000 hand super-pixels and 80000 background super-pixels. After that, we generate different versions of hand map proposals by applying the thresholded Grabcut [35] algorithm. Namely, we apply the Grabcut onto the hand detection score maps with different parameter settings w.r.t. to sure foreground, sure background, possible foreground, and possible background. A set of N_s ($N_s = 24$) proposals (denoted as a proposal set $\mathcal{P}_n$, $|\mathcal{P}_n| = N_s$) are generated for each weakly supervised image I_n.

To seek the best hand segmentation hypothesis s_n for each weakly supervised hand bounding box image I_n, we describe the measurement criteria (indication of a *good hypothesis*) as:

$$\varepsilon_s(\theta, s_n) = \frac{1}{N}\sum_{n=1}^{N}\left(1 - \kappa\left(h(I_n|\theta), l\left(I_n|s_n\right)\right)\right), s_n \in \mathcal{P}_n, \tag{1}$$

where $h(I_n|\theta)$ is the pixel-level prediction for image I_n from the hand segmentation DCNN h with parameters θ, $l\left(I_n|s_n\right)$ is the pixel-level hand segmentation map from the hand map proposal s_n, $s_n \in \mathcal{P}_n$. To compute the overlap ratio between $h(I_n|\theta)$ and $l\left(I_n|s_n\right)$, we define $\kappa\left(h(I_n|\theta), l(I_n|s_n)\right)$ as:

$$\kappa\left(h(I_n|\theta), l(I_n|s_n)\right) = \frac{h(I_n|\theta)\bigcap l(I_n|s_n)}{h(I_n|\theta)\bigcup(I_n|s_n)}, \tag{2}$$

which is the intersection over union ratio of the predicted and ground-truth hand regions. For the predicted map, we threshold it into a binary map by the resulting hand probability scores, the threshold value is 0.5. Equation 1 is normalized by the number of images N.

Intuitively, we might pick up the best proposal s_n for each hand bounding box image I_n, *i.e.*, to achieve the largest $\kappa\left(h(I_n|\theta), l(I_n|s_n)\right)$. However, this greedy solution may be trapped to a bad local optima, by always picking up the same candidate from the hand map proposal set $\mathcal{P}_n$. To overcome this issue, we randomly select one of the best K ($K = 3$) hand segmentation hypotheses, instead of always picking up the best one.

Algorithm 1 pixel-level hand detection algorithm

Input: Hand bounding box images $\{I_n\}_{n=1:N}$, initialized network $h(I_n|\theta)$ with the parameter set θ.
Output: $h(I_n|\theta)$
(1) Apply super-pixel based image segmentation on each hand bounding box image I_n to generate hand probability map $I_n^{'}$.
(2) Apply the thresholded Grabcut on each $I_n^{'}$ to generate a set of hand segmentation proposals $\mathcal{P}_n$, each hand map hypothesis is denoted as $l(I_n^{'}|s_n)$.
(3) EM-training algorithm:
$i = 1$, $N_{max} = 10$
while $i \leq N_{max}$ **do**
  E-step: fix θ, optimize s_n in Equation 4. *i.e.*, for each image, select candidate $s_n \in \mathcal{P}_n$ to achieve the largest κ, in Equation 1.
  M-step: fix the set $\{s_n\}$, optimize θ in Equation 4. *i.e.*, apply SGD training to update θ, in Equation 3.
  $i = i + 1$
end while
(4) Infer each hand bounding box image I_n with the network $h(I_n|\theta)$

M-Step: Refine the Hand Segmentation Network. With the augmented training set, *i.e.*, the weakly supervised hand bounding boxes and the inferred best hand map hypotheses, we fine-tune the hand segmentation network $h(I_n|\theta)$. We formulate the objective of the network training as a per-pixel regression problem to the selected hand map proposals $\{s_n\}$ from E-step. More formally, the objective function is written as:

$$\varepsilon_\theta(\theta, s_n) = e\left(h(I_n|\theta), l(I_n|s_n)\right), \tag{3}$$

where $e\left(h(I_n|\theta), l(I_n|s_n)\right)$ is the cross-entropy error function.

The ultimate objective function is concluded as the combination of Equation 1 and Equation 3:

$$\min_{\theta,\{s_n\}} \sum_{n=1}^{N}\varepsilon_s(\theta, s_n) + \lambda\sum_{n=1}^{N}\varepsilon_\theta(\theta, s_n), \tag{4}$$

where λ is the weight parameter, which we fix it as 3 by cross validation. To minimize the objective function, the network parameter θ and the best hand map hypotheses $\{s_n\}$ are alternatively optimized. The procedure of the hand segmentation network training can be solved by the back propagation and stochastic gradient descent (SGD). In the SGD training, we use a mini-batch size of 8. The learning rate and momentum are initialized to be 0.001 and 0.9, learning rate is divided by 10 after every 4K iterations, the

weight decay is set as 0.0005. The training procedure lasts for 12K SGD iterations. This is the training parameters for one epoch of our iterative training. We keep the training for N_{max} ($N_{max} = 10$) epochs.

3.2. Foreground/Active Object Region Localization

On one hand, we note that precise detection of **hands** could help localize the foreground manipulated objects in some actions. For example, in the actions such as "operating the TV monitor", "pouring water with bottle", "turning on the tap", "drink water/bottle", *etc*., hands are always very close to the objects-of-interest. Therefore, to find the hands directly secures the object locations. On the other hand, we also note that sometimes **motion** is also a good indication of the location of the foreground object, *i.e*., foreground object movements are different from background motions. For example, the movements of washing hands are significantly more consistent and stronger than the background motions. We note that this is the first time that hand map, motion map and object map are combined together to generate foreground interactional object regions, in an end-to-end network. The advantage of the network is that rich mutual contextual information could be explored.

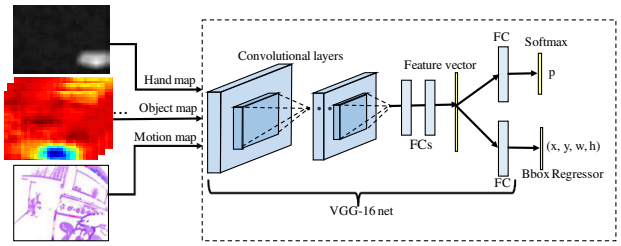


Figure 3: The deep bounding box regression neural network for manipulated foreground object detection.

Network Architecture. Inspired by these observations, we employ an end-to-end DCNN to take the corresponding hand map with, object feature map, and motion map to detect the active object region, *i.e*., interactional foreground. Firstly, the hand maps (*i.e*., binary masks) are generated from the above introduced pixel-level hand detection/segmentation network. Secondly, we calculate the optical flow motion map of the original image, from two sequential video frames. Thirdly, we input the training object bounding box images to the AlexNet [18] image classification deep network. For each image, we extract the top-5 strongest object feature maps (the ones with strongest responses) of the fifth convolutional layer (Conv_5) as the object map. The three maps with the raw image are fed into a bounding box regression network based on the VGG-16 net, as illustrated in Figure 3. The deep network mainly contains 16 weights layers, 13 convolutional layers, 3 fully connected layers, softmax layer and bounding box regression layer. The bounding box regressor is used to predict the object location and the softmax classifier outputs the object class probability. This regression network prototype can be found in the work [10]. For each image, the outputs from the network are the tuples of (x, y, w, h, p), where (x, y, w, h) is the object location for the detected bounding box, and p is the corresponding object class probability.

Network Training. The two objectives of network training are to predict object bounding boxes and their confidence scores for each training image, such that the highest scoring boxes well match the ground-truth bounding boxes for the image, both objectives can be jointly modeled as in [10]. The ground truth for passive and active bounding boxes are provided by the dataset ADL [31]. To train the network, we firstly fine-tune all the convolutional layer weights by the pre-trained VGG-16 model on ImageNet. We initialize learning rate as 0.001 and run SGD for 30k mini-batch iterations, then lower the learning rate to 0.0001 and train for another 10k iterations. A momentum term with weight 0.9 and weight decay factor of 0.0005 are used in the experiments. We regard the trained model as the **active object model**. In our work, we also train the **passive object model** by feeding the bounding box regression network [10] with only passive object bounding boxes, without using the active object detection.

3.3. Action Representation

We include both key **object** information and **motion** information in our action representation framework. On one hand, we follow [31] to compute the object features from the detected object locations and confidence scores, then we represent the object features in a temporal pyramid manner [31], and obtain a global object representation for each action. In section 3.2, we train both active object model (*i.e*., to detect manipulated foreground objects) and passive object model (*i.e*., without considering active objects), thus each action is formed into the active object histogram and passive object histogram. On the other hand, we also propose to utilize state-of-the-art improved dense trajectories [44] to represent motion characteristics in egocentric actions. For each trajectory, we extract the motion features including HOG (96-dim), and MBH (96-dim for MBHx, 96-dim for MBHy), they are reduced to 64-dim by PCA. We train Gaussian mixture models with 64 components, and encode each action with the improved Fisher vectors [30] (8192-dim, $64 \times 64 \times 2$), we regard the trained model as **global motion pooling**. To apply **local motion pooling**, we perform the feature encoding and pooling from the trajectories that are going through the hand regions or the interactional foreground objects, note that we regard the top-scored bounding box as the active object for each image.

To this end, we obtain both object model (passive+active object model) and motion model (global+local motion pooling). To combine the representations from each feature channel, we adopt a multi-channel approach [43]. Based

on the combined feature channel mapping, we train a non-linear SVM classifier. We fix the regularization parameter C=10 by cross validation. We use the LibSVM [3] as our SVM solver.

4. Experiment

We evaluate our hand segmentation method on the Georgia Tech Egocentric Activity (GTEA) dataset [9], and we evaluate the action recognition framework on the egocentric video benchmark Activities of Daily Living (ADL) [31].

4.1. Hand Segmentation Performance

Dataset. The GTEA dataset [9] contains 7 types of daily activities, each is performed by 4 different subjects. The tested frames are taken from the actions of subject one, who is making tea, making a peanut butter sandwich or making coffee. Follow the same settings in [9], we use the coffee sequence as training when testing on the tea and peanut sequence, and we use the tea sequence as training when testing on the coffee sequence. We use the F-score (*i.e.*, harmonic mean of the precision and recall) to quantitatively evaluate the segmentation performance. The scores are computed by comparing the predictions (*i.e.*, segmentation) to the ground truth from the project site.

Table 1: Hand segmentation performance comparison on GTEA dataset.

Method	peanut	coffee	tea
Trajectory projection [39]	0.255	0.275	0.239
Single pixel color [13]	0.730	0.837	0.804
Superpixel + CRF [9]	0.727	0.713	0.812
Global scene [20]	0.883	0.933	0.943
Proposed	**0.912**	**0.954**	**0.962**

Table 1 shows the comparative hand segmentation results on the GTEA dataset. Firstly, the super-pixel [9] + CRF and Single pixel color [13] methods are based on the effective low-level features in the hand segmentation task. Secondly, the Global scene method [20] integrates different low-level features (*e.g.*, color feature or texture feature), and models the global background scenes (*i.e.*, to mitigate the large illumination changes), to achieve better performance than the previous methods. However, the performance is sensitive to the number of scene categories, and it is extremely difficult to model the scenes for the large-scale dataset in real life. Instead, our method can directly learn rich features from the neural network by feeding the network with various training data that are under different scenes. We observe that our performance outperforms all the above methods.

In Figure 4, we show that both of the Global scene method [20] and our proposed method can perform well under the controlled environment. However, it is indeed difficult to model all the background scenes, we show such difficulties for Global scene method in row one of Figure 5,

Figure 4: Successful hand segmentation examples of Global scene method [20] (row one) and our method (row two).

Figure 5: Hand segmentation comparison between Global scene method [20] (row one) and our method (row two) under background scene changes.

Figure 6: Some failure examples of our method under extreme darkness or extreme brightness.

and it is expected that these failures can lead to rather unstable performance. We show that our method is still effective compared to the Global scene method, in row two of Figure 5. However, it is intuitive to expect rather bad performance under extreme darkness or extreme brightness. We show such failure cases of our proposed method in Figure 6.

4.2. Acton Recognition Performance

Dataset. The ADL [31] dataset consists of 20 egocentric videos which are collected by 20 persons. Both action annotations (*i.e.*, start time, end time, and action label for each video sequence) and object annotations (*i.e.*, object class, object bounding box, passive/active status) are provided. A total of 18 action categories and 44 objects are annotated. During object detection, the actions of the first 6 videos are used as training and the rest are used for testing. To evaluate the action classification performance, we perform the leave-one-person-out cross validation, We report the per-class average precision (mAP) score, with equal weight for each action class.

First of all, we define the following methodology terms: 1) **Passive object** trains the object detection DCNN by only using the passive object bounding box data, without active object detection; 2) **Passive+Active object** augments the passive object model by combining the passive object histogram with the active object histogram from the active object detection; 3) **Global motion pooling** uses the recent

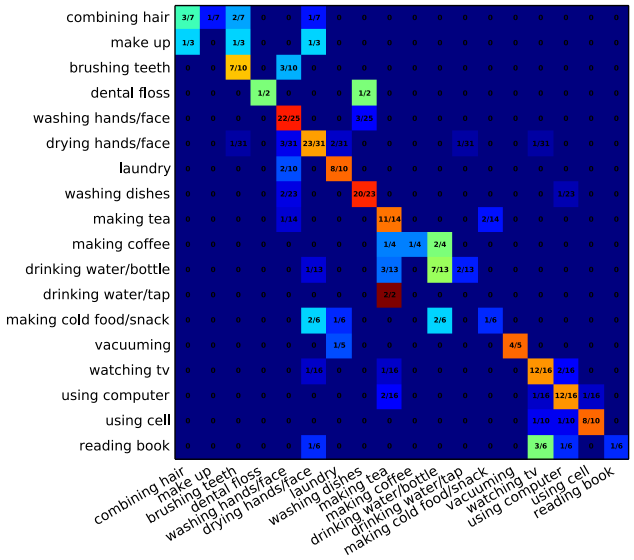


Figure 7: Confusion matrix resulted from the combined Object + Motion model in Table 2.

popular improved dense trajectories [44]; 4) **Global+Local motion pooling** combines the Global motion pooling with Local motion pooling on the active foreground regions; 5) **Object+Motion** is the model to combine **Passive+Active object** and **Global+Local motion pooling**.

Table 2: Action classification performance comparison on ADL dataset.

Method	mAP (%)
BoW	16.5
Boost-RSTP [25]	33.7
Boost-RSTP + OCC [25]	38.7
Bag-of-objects [31]	32.7
Bag-of-objects + Active model [31]	36.9
Passive object	**35.2**
Passive+Active object	**43.8**
Global motion pooling	**36.7**
Global+Local motion pooling	**42.5**
Object + Motion	**55.2**

In Table 2, we compare our approach with the state-of-the-art methods on ADL dataset. Firstly, we show that the motion-based **Global motion pooling** is very useful in egocentric action representation. Secondly, the object-centric methods, such as Bag-of-objects, Boost-RSTP Boost-RSTP + OCC, and Bag-of-objects+Active model, combine the passive and active object features with temporal pyramid, and can achieve comparative performance to the **Global motion pooling** method. However, none of the above methods can well solve the active object detection problem. We improve these methods by jointly modeling the object, hand and motion information from the deep neural network. We show that our **Passive+Active object** model has outperformed the previous object-centric methods. We also augment the **Global motion pooling** with Local motion pooling, by highlighting active foreground regions (objects-of-interest) and suppressing the background noise. Finally, we combine the complementary object model and motion model (**Object + Motion** In Table 2), to achieve the aggregated performance, we also show its confusion matrix in Figure 7. Particularly, we find our method can have good accuracy for those manipulation actions such as watching tv (operating tv remote), using computer (typing keyboard), washing hands (manipulating hands, tap), *etc*.

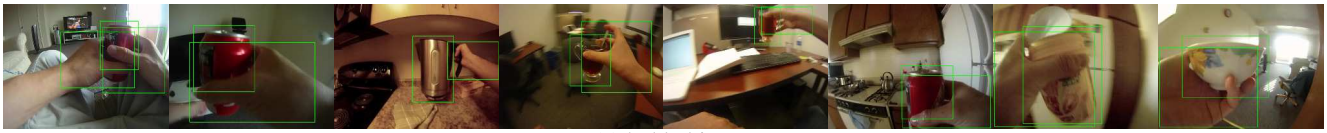
Table 3: Correlation between different hand segmentation accuracy and action recognition performance (mAP).

	mAP (%)		
Method	Passive+Active object	Global+Local motion pooling	Object + Motion
Bounding box	38.5	39.8	48.1
Superpixel + CRF	39.3	40.1	49.8
Global scene	33.7	38.2	44.3
Proposed	**43.8**	**42.5**	**55.2**

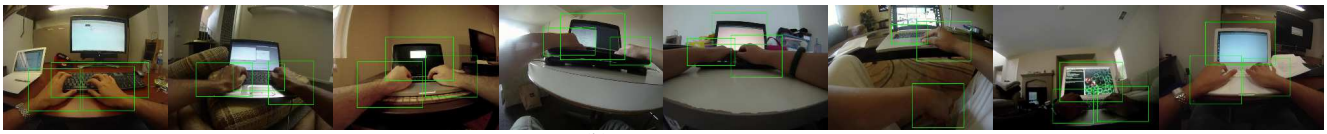
In Table 3, we show the correlation between hand segmentation accuracy and action recognition performance. Firstly, we show the largely degenerated performance by replacing the hand segmentation maps with hand bounding boxes (first method in Table 3). Secondly, the other hand segmentation methods such as Superpixel + CRF [9] and Global scene [20] intuitively result in worse action recognition performance because their hand segmentation accuracy cannot compare to ours. We observe that the Global scene method achieves even worse performance than the bounding box input, as we find it is extremely difficult to model all the scene categories by using the simple scene clustering algorithm. In contrast, our hand segmentation method can directly learn various background scenes from the deep neural network, and results in the best action recognition performance in Table 3.

In Figure 9, we show that the presence of hand, object and motion maps can influence training of the active object detection network CNN2, therefore affecting the ultimate action recognition performance. We use the **Passive+Active object** method for performance evaluation. We show that all types of maps are necessary for performance improvement, specifically, the hand map and object map are most significant of all. We also expect that motion map cannot be compared to the other two maps because of the background noise.

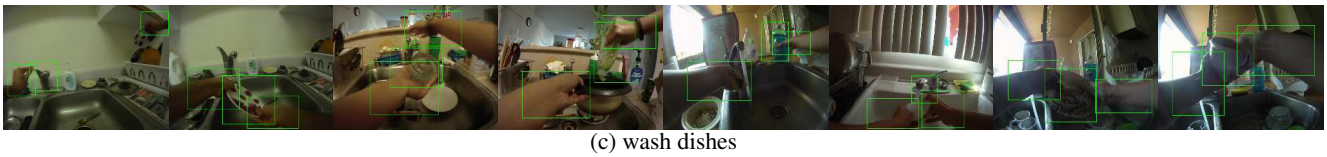
We list some exemplars of active foreground regions from the active object detection neural network CNN2, in Figure 8. These detected objects are either manipulated by hands (*e.g.*, keyboard, kettle, tv remote) or nearby (*e.g.*,



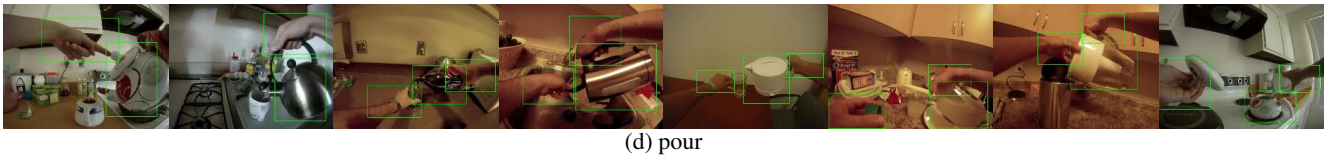
(a) hold object



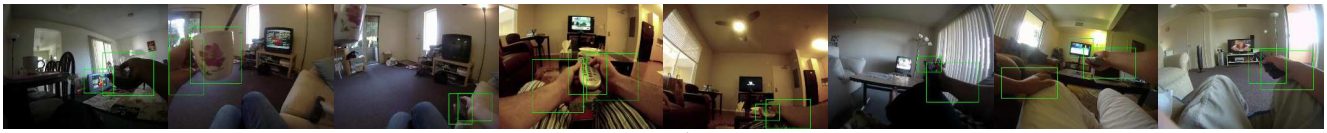
(b) use computer



(c) wash dishes



(d) pour



(e) watch tv

Figure 8: The active foreground regions detected from the active object detection neural network CNN2. The green bounding boxes consist of the hands and co-localized/detected manipulated objects, for different action categories under various scenes. For each image, we visualize the active object as the bounding box to achieve the topest confidence score.

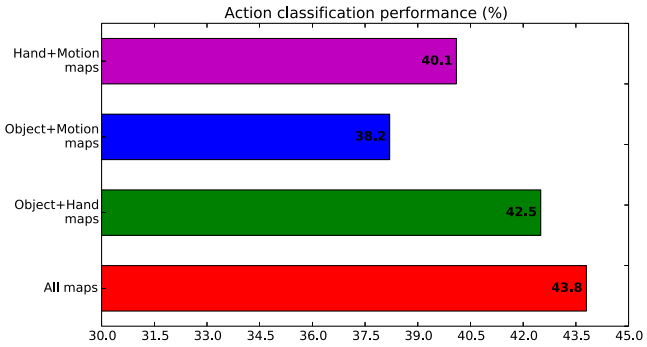


Figure 9: Correlation between action recognition performance and presence of hand, object and motion map in training the active object detection network CNN2.

monitor, laptop, detergent). They are easier for detection because finding the hands can co-localize the active objects.

All the experiments are conducted on a computing server with two Intel Xeon E5450 Quad Core processors (3.00GHz) and 32 GB memory, the computational platform is equipped with one Nvidia Tesla K40 GPU. The deep semantic parsing neural network CNN1 is based on the DeconvNet package [28], total of 24,569 hand bounding box images are used for the training. The network training speed is 5 seconds per iteration, it takes approximately 14 hours for each epoch of our iterative EM-like training algorithm, and 6 days to finish the training procedure. The testing speed is 1.35 seconds/image. The active object detection neural network CNN2 is based on the Fast-rcnn [10] package. We sample one video frame every second for object detection. The training time is 13 hours in total for 11,643 images (*i.e.*, 6 out of 20 subjects are used for training). The prediction time is even faster, *i.e.*, 0.2 second per image.

5. Conclusion

Firstly, we propose a novel pixel-to-pixel deep convolutional neural network to achieve decent hand segmentation performance. Secondly, the resulting hand maps are further paired with motion maps and object maps via another object detection DCNN, which explores the contexts among object, motion and hand to generate foreground interactional objects. Experiments show that our framework has achieved the state-of-the-art egocentric action recognition performance on the benchmark dataset ADL.

Acknowledgments.

This work was partially supported by National Natural Science Foundation of China (61502301 and 61429201). This work was also partially supported to Dr. Qi Tian by ARO grants W911NF-15-1-0290 and Faculty Research Gift Awards by NEC Laboratories of America and Blippar.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *T-PAMI*, 34(11):2274–2282, 2012.

[2] A. A. Argyros and M. I. Lourakis. Real-time tracking of multiple skin-colored objects with a possibly moving camera. In *ECCV*, pages 368–379. Springer, 2004.

[3] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[5] A. Y. Dawod, J. Abdullah, and M. J. Alam. Adaptive skin color model for hand segmentation. In *Computer Applications and Industrial Electronics (ICCAIE), 2010 International Conference on*, pages 486–489. IEEE, 2010.

[6] A. Fathi, A. Farhadi, and J. M. Rehg. Understanding egocentric activities. In *ICCV*, pages 407–414. IEEE, 2011.

[7] A. Fathi, J. K. Hodgins, and J. M. Rehg. Social interactions: A first-person perspective. In *CVPR*, pages 1226–1233. IEEE, 2012.

[8] A. Fathi, Y. Li, and J. M. Rehg. Learning to recognize daily actions using gaze. In *ECCV*, pages 314–327. Springer, 2012.

[9] A. Fathi, X. Ren, and J. M. Rehg. Learning to recognize objects in egocentric activities. In *CVPR*, pages 3281–3288. IEEE, 2011.

[10] R. Girshick. Fast r-cnn. *arXiv preprint arXiv:1504.08083*, 2015.

[11] I. González Díaz, V. Buso, J. Benois-Pineau, G. Bourmaud, and R. Megret. Modeling instrumental activities of daily living in egocentric vision as sequences of active objects and context for alzheimer disease research. In *Proceedings of the 1st ACM international workshop on Multimedia indexing and information retrieval for healthcare*, pages 11–14. ACM, 2013.

[12] E. Hayman and J.-O. Eklundh. Statistical background subtraction for a mobile observer. In *CVPR*, pages 67–74. IEEE, 2003.

[13] M. J. Jones and J. M. Rehg. Statistical color models with application to skin detection. *IJCV*, 46(1):81–96, 2002.

[14] P. Kakumanu, S. Makrogiannis, and N. Bourbakis. A survey of skin-color modeling and detection methods. *Pattern recognition*, 40(3):1106–1122, 2007.

[15] M. Kölsch and M. Turk. Analysis of rotational robustness of hand detection with a viola-jones detector. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 107–110. IEEE, 2004.

[16] M. Kölsch and M. Turk. Robust hand detection. In *FGR*, pages 614–619, 2004.

[17] M. Kölsch and M. Turk. Hand tracking with flocks of features. In *CVPR*, volume 2, pages 1187–vol. IEEE, 2005.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[19] C. Li and K. Kitani. Model recommendation with virtual probes for egocentric hand detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2624–2631, 2013.

[20] C. Li and K. M. Kitani. Pixel-level hand detection in egocentric videos. In *CVPR*, pages 3570–3577. IEEE, 2013.

[21] Y. Li, A. Fathi, and J. M. Rehg. Learning to predict gaze in egocentric video. In *ICCV*, pages 3216–3223. IEEE, 2013.

[22] Y. Li, Z. Ye, and J. M. Rehg. Delving into egocentric actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 287–295, 2015.

[23] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos "in the wild". In *CVPR*, pages 1996–2003. IEEE, 2009.

[24] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *arXiv preprint arXiv:1411.4038*, 2014.

[25] T. McCandless and K. Grauman. Object-centric spatio-temporal pyramids for egocentric activity recognition. In *BMVC*. Citeseer, 2013.

[26] A. Mittal, A. Zisserman, and P. H. Torr. Hand detection using multiple proposals. In *BMVC*, pages 1–11. Citeseer, 2011.

[27] S. Narayan, M. S. Kankanhalli, and K. R. Ramakrishnan. Action and interaction recognition in first-person videos. In *CVPRW*, pages 526–532. IEEE, 2014.

[28] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. *arXiv preprint arXiv:1505.04366*, 2015.

[29] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Markerless and efficient 26-dof hand pose recovery. In *ACCV*, pages 744–757. Springer, 2011.

[30] D. Oneata, J. Verbeek, and C. Schmid. Action and event recognition with fisher vectors on a compact feature set. In *ICCV*, pages 1817–1824. IEEE, 2013.

[31] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *CVPR*, pages 2847–2854. IEEE, 2012.

[32] Y. Poleg, C. Arora, and S. Peleg. Temporal segmentation of egocentric videos. In *CVPR*, pages 2537–2544. IEEE, 2014.

[33] J. M. Rehg and T. Kanade. Visual tracking of high dof articulated structures: an application to human hand tracking. In *ECCV*, pages 35–46. Springer, 1994.

[34] X. Ren and M. Philipose. Egocentric recognition of handled objects: Benchmark and analysis. In *CVPRW*, pages 1–8. IEEE, 2009.

[35] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)*, 23(3):309–314, 2004.

[36] M. S. Ryoo and L. Matthies. First-person activity recognition: What are they doing to me? In *CVPR*, pages 2730–2737. IEEE, 2013.

[37] M. S. Ryoo, B. Rothrock, and L. Matthies. Pooled motion features for first-person videos. *arXiv preprint arXiv:1412.6505*, 2014.

[38] A. J. Sellen, A. Fogg, M. Aitken, S. Hodges, C. Rother, and K. Wood. Do life-logging technologies support memory for the past?: an experimental study using sensecam. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 81–90. ACM, 2007.

[39] Y. Sheikh, O. Javed, and T. Kanade. Background subtraction for freely moving cameras. In *ICCV*, pages 1219–1225. IEEE, 2009.

[40] L. Sigal, S. Sclaroff, and V. Athitsos. Skin color-based video segmentation under time-varying illumination. *T-PAMI*, 26(7):862–877, 2004.

[41] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[42] B. Stenger, P. R. Mendonça, and R. Cipolla. Model-based 3d tracking of an articulated hand. In *CVPR*, volume 2, pages II–310. IEEE, 2001.

[43] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, pages 3169–3176. IEEE, 2011.

[44] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, pages 3551–3558. IEEE, 2013.

[45] Y. Zhou, B. Ni, R. Hong, M. Wang, and Q. Tian. Interaction part mining: A mid-level approach for fine-grained action recognition. In *CVPR*, pages 3323–3331, 2015.