Progressively Parsing Interactional Objects for Fine Grained Action Detection

Bingbing Ni
Shanghai Jiaotong University
Shanghai, China
nibingbing@sjtu.edu.cn

Xiaokang Yang
Shanghai Jiaotong University
Shanghai, China
xkyang@sjtu.edu.cn

Shenghua Gao
ShanghaiTech University
Shanghai, China
gaoshh@shanghaitech.edu.cn

Abstract

Fine grained video action analysis often requires reliable detection and tracking of various interacting objects and human body parts, denoted as Interactional Object Parsing. *However, most of the previous methods based on either independent or joint object detection might suffer from high model complexity and challenging image content, e.g., illumination/pose/appearance/scale variation, motion, and occlusion etc. In this work, we propose an end-to-end system based on recurrent neural network to perform frame by frame interactional object parsing, which can alleviate the difficulty through an incremental/progressive manner. Our key innovation is that: instead of jointly outputting all object detections at once, for each frame we use a set of long-short term memory (LSTM) nodes to incrementally* refine *the detections. After passing through each LSTM node, more object detections are* consolidated *and thus more contextual information could be utilized to localize* more difficult *objects. The object parsing results are further utilized to form object specific action representation for fine grained action detection. Extensive experiments on two benchmark fine grained activity datasets demonstrate that our proposed algorithm achieves better interacting object detection performance, which in turn boosts the action recognition performance over the state-of-the-art.*

1. Introduction

Fine-grained action analysis has been an emerging research direction during recent years [22, 14, 20, 18, 37, 36]. The major task of fine-grained interaction action analysis is to detect the interacting objects or human body parts for each video frame (in the rest of the paper, we will simply use the term *objects* to denote both interactional objects and human body parts). However, detecting and tracking the objects under interaction (also denoted as *interactional object parsing*) in fine-grained action videos is very challenging due to frequent occlussion/self-occlusion, change of object scale/orientation/appearance, and fast object or background motion, etc.

Early methods detect and track each of the interacting objects **independently**. For example, some works [3, 27, 15] only attempt to detect and track hands in interaction videos, without consideration of the relationship between the hands and the objects being manipulated. In some contextual object recognition works [17, 7, 33, 11], hands and objects are detected using different methods independently, and the detections are further fused via probabilistic graphical models for high level inference, e.g., action detection or joint object and action recognition. With RGB-Depth data, Lei et al. [14] proposed a system to perform fine-grained kitchen activity recognition. Different objects are detected independently. Obviously, this method is not able to utilize rich contextual information among various objects during interaction, e.g., hands and objects, to improve the joint detection performance.

Geometrical contextual information among objects, body parts and body poses could be **jointly** explored to enhance both object and action recognition performance. Packer et al. [20] presented a joint model for objects, human poses and motion features to recognize complex, fine-grained human actions in cooking action sequences. Koppula et al. [12] jointly detected manipulation activities and object affordances. These works require 3D skeleton data which are not easy to obtain in practice. Ni et al. [18] recently proposed a joint hand and object tracking framework called *interaction tracking*, which is based on the observation that there exists rich contextual information between the interaction status and the occurrence of mutual occlusion. Their method outperforms prior art.

However, joint object detection framework still has two issues. First, joint object detection frameworks are usually based on tree-like graphical models, for example, deformable part-based models [5, 32], And-Or graph models [38, 16], probabilistic graphical models [20, 18]. These models usually can only handle the binary contextual relationship between objects. While a large portion of actions involve high order interaction, e.g., hand, knife, and chopping board for the action *cutting*, simply considering the mutual geometrical relationship between objects is insufficient to guarantee good object parsing performance. It is infeasible for these algorithms to model higher order contextual relationship due to the high computational complexity nature of these algorithms. Therefore, we need a flexible way to model high order contextual information for better detection of interactional objects. Second, during fine grained interaction there exist frequent occlusion/self-occlusion and object appearance/scale/orientation change, which make the interactional object parsing problem very challenging. Therefore, it is very hard to guarantee the performance of joint object detection [18]. One key observation is that during interaction, some objects might be ”easier” to detect than others. For example, during the *chopping* action, the chopping board and the hand is often easier to detect than the knife (which is often occluded). If we can confidently identify some ”easy” objects first, it would be very helpful for us to further use the contextual information to seek other related and more ”difficult” objects. In other words, an easy-to-difficult progressive/incremental detection approach might be more preferable for interactional object parsing. Moreover, the detection order should be varying according to different interaction scenarios.

Inspired by the recent success of recurrent neural networks (RNN) [6] (especially application of the long-short term memory network (LSTM) [8] in people detection [26]), we propose an end-to-end system based on recurrent neural network to perform frame by frame interactional object parsing, which can alleviate the above mentioned difficulty through a progressive/incremental detection manner. For each frame, expressive image features from the state-of-the-art deep convolutional models (e.g., VGG-19 [25]) are input to our proposed interactional object parsing network. Instead of jointly outputting all object detections at once, for each frame, we use a set of LSTM nodes to progressively/incrementally *refine* the detections. After passing through each LSTM node, more object detections are *consolidated* and thus more contextual information could be utilized to determine *more difficult* object detections. By applying the proposed network, all detection and contextual information up to the current nodes could be maximally explored to generating a better detection in the next processing LSTM node. Therefore, ”easy” objects have higher probability to be confidently detected and confirmed in the early LSTM nodes of the network. Based on the contextual information between these already ”discovered” objects with the uncertain ones, the later LSTM nodes could better identify the ”more difficult” objects (e.g., those occluded ones during interaction, or those have large appearance/scale/orientation variation). The detection results of the current frame are also input to the LSTM network associated with the next frame to facilitate inter-frame tracking.

The object parsing results are further used to compute object specific motion representations for fine grained action detection. We perform extensive experiments on two benchmark fine grained activity datasets. The results demonstrate that our proposed algorithm achieves better interacting object detection performance, which in turn boosts the action recognition performance over the state-of-the-art.

The rest of this paper is organized as follows. We enumerate some related works in Section 2. The details of our progressive interactional object parsing network along with the detailed implementation and training procedure are described in Section 3. We demonstrate experimental settings and results in Section 5. Conclusions are given in Section 6.

2. Related Works

Fine-grained Action Analysis. Although general action recognition has a rich literature [19, 13, 23, 30, 35], fine-grained action analysis is a relatively new research direction. Rohrbach et al. [22] provided a large-scale fine-grained cooking action dataset with several baseline results. Their dataset has been the most important and challenging benchmark test-bed for evaluating fine-grained action recognition/detection algorithms. Several works [14, 20] utilized RGB-depth information for fine-grained action recognition; however, relying depth channel is a limitation for realistic application. Zhou et al. [37, 36] proposed a series of works on fine-grained action analysis which focus on modeling local contextual information between motion and object of interest. The work mostly related to ours is by Ni. et al. [18], where their focus is also object parsing/tracking in fine-grained action video.

Recurrent Neural Networks and LSTM. Recurrent neural networks especially the long-short term memory models [8] have achieved great success in a large variety of applications including temporal modeling such as natural language processing [24] and speech recognition [6], and non-temporal modeling such as image caption generation [10, 29].

Several works have been proposed to model action image sequences using RNN/LSTM models. Veeriah et al. [28] proposed a differential gating scheme for the LSTM neural network (termed as differential Recurrent Neural Network (dRNN)), which emphasizes on the change in information gain caused by the salient motions between the successive frames. Donahue et al. [4] developed a novel recurrent convolutional architecture for large-scale visual learning. They applied this model on several tasks including benchmark video recognition, image description, and video narration. Wu et al. [34] extracted spatial and the short-term motion features by two Convolutional Neural Networks (CNN) to further model longer-term temporal clues. The two types of CNN-based features are further combined in a regularized feature fusion network for video event classification. The above works mostly focus on modeling temporal dependencies for action recognition; in contrast, our work focuses on progressive refinement of the interactional object detection within each video frame, i.e., the LSTM model applied to our problem is not focused on temporal modeling, rather, it is applied for sequencing the detection process in an easy-to-difficult manner. In this sense, the most related work to ours is by Stewart and Andriluka al. [26], where LSTM network is applied for sequencing the human detection problem. Namely, for each frame, their model sequentially outputs the detection bounding boxes by exploring the contextual information between bounding boxes. Their algorithm, however, is designed for detecting only a single type of object (with many instances in the image). In contrast, our algorithm handles different types of objects simultaneously, based on our developed sequential detection refinement algorithm.

3. Interactional Object Parsing Network

Motivation. The task of interactional object parsing is to infer at each frame the bounding boxes for various interacting objects and human parts. To this end, we propose an interactional object detection network which progressively/incrementally outputs/refines the bounding boxes for various interactional objects at each frame. In other words, instead of confirming the localizations of all the object bounding boxes simultaneously at each frame, the image frame is input to a recurrent neural network and that network progressively refines the object detection results node-by-node until all detected object bounding boxes cannot be improved further (e.g., when the detection confidence is below some threshold). Through this progressive detection scheme, objects that are ”easy” to detect (e.g., without occlusion, with little deformation etc.) could be identified and consolidated in the early output nodes of the recurrent neural network. The detections which are ”consolidated” from early nodes of the recurrent network could provide contextual information, which help to detect more ”difficult” objects (e.g., those occluded or deformed ones) in later output nodes of the network. For instance, during a *frying* action, the hand and the fry pan might be detected much easier and their confirmed positions could help to locate the position of the ladder, which is most probably occluded by hand and difficult to localize.

Network Architecture. We begin with notations. The number of interesting objects (including human body parts, e.g., hands) is denoted as M. At each frame, our parsing network outputs a concatenated vector $\mathbf{B}$, which is composed of M object bounding boxes with detection confidence scores $\mathbf{B} = (\mathbf{b}_1; \mathbf{b}_2; \cdots; \mathbf{b}_M)$. Here each $\mathbf{b}_m = (\mathbf{p}_m; c_m)$ is composed of four dimensional bounding box parameters $\mathbf{p}_m$ indicating the relative x-y coordinates and height, width of the bounding box for object m, as well as the corresponding detection confidence score c_m. Higher value of the confidence score indicates higher probability that the detected bounding box matches the target object. Figure 1 overviews our proposed end-to-end interactional object parsing network. Each frame image is input to a VGG-19 [25] CNN model for extracting image representation. As the original image frame size is usually $640 \times 480 \times 3$, we first re-scale it to $320 \times 240 \times 3$ and we crop the center $224 \times 224 \times 3$ region for processing. Note that in fine grained action, the human subject along with the objects being manipulated are always in the image center. The $Conv_5$ layer ($D = w \times h \times 512$-dimensional, w and h denote the receptive field size) is used as the image level representation. We denote this feature as CNN feature $\mathbf{x}$. We use the VGG-19 network architecture and its ImageNet pre-trained model due to its discriminative capability in image classification task. Note that each pixel in the $Conv_5$ map has the receptive field size typically smaller than that of any object of interest. Namely, the *resolution* of the $Conv_5$ CNN feature map $\mathbf{x}$ is sufficient for object localization. The D-dimensional CNN image representation is further input to a recurrent network structure with H LSTM nodes (in this work, H is larger than M, i.e., to allow sufficient number of iterative refinement steps). Each LSTM takes the D-dimensional CNN image representation and the object parsing status $\mathbf{C}^{(l-1)}$ vector from the last LSTM node and outputs the detection vector $\mathbf{B}^{(l)}$ for the current LSTM node (we use l to index the LSTM node). We set the dimensionality of LSTM cell status vector $\mathbf{C}^{(l)}$ as 512. In other words, at each frame t, our LSTM network generates a sequence of gradually refined object parsing vectors, i.e., $\mathbf{B}_t^{(1)}, \mathbf{B}_t^{(2)}, \cdots, \mathbf{B}_t^{(H)}$. Inter-frame tracking is naturally handled by inputting the cell status of the last LSTM node ($l = H$) of the current frame t to the first LSTM node ($l = 1$) of the next frame $t + 1$. This interactional object parsing process for frame t could be mathematically expressed as:

$$\begin{aligned}
\mathbf{i}_l &= \sigma\big(W_i \mathbf{x}_t + U_i \mathbf{h}_t^{(l-1)} + \mathbf{b}_i\big),\\
\mathbf{f}_l &= \sigma\big(W_f \mathbf{x}_t + U_f \mathbf{h}_t^{(l-1)} + \mathbf{b}_f\big),\\
\mathbf{o}_l &= \sigma\big(W_o \mathbf{x}_t + U_o \mathbf{h}_t^{(l-1)} + \mathbf{b}_o\big),\\
\widetilde{\mathbf{C}}_t^{(l)} &= \tanh\big(W_c \mathbf{x}_t + U_c \mathbf{h}_t^{(l-1)} + \mathbf{b}_c\big),\\
\mathbf{C}_t^{(l)} &= \mathbf{i}_l * \widetilde{\mathbf{C}}_t^{(l)} + \mathbf{f}_l * \widetilde{\mathbf{C}}_t^{(l-1)},\\
\mathbf{h}_t^{(l)} &= \mathbf{o}_l * \tanh(\mathbf{C}_t^{(l)}),\\
\mathbf{B}_t^{(l)} &= softmax(W \mathbf{h}_t^{(l)} + \mathbf{b}). \qquad (1)
\end{aligned}$$

Here $\mathbf{i}_l$, $\mathbf{f}_l$, $\mathbf{o}_l$ and $\mathbf{C}^{(l)}$ denote the input gate, forget gate, output gate, and cell status of the LSTM node l, respectively. Note that for each frame, the image CNN feature $\mathbf{x}_t$ is input to all LSTM nodes. Through this recursive architecture, image features and contextual information (e.g., early consolidated detections) could be jointly explored to gradually detect more and more ”difficult” objects.

Cost Function. To facilitate our idea of progressive refinement of interactional object detections, we introduce a *partial matching* cost, mathematically defined as follows:

$$\begin{aligned}\ell_{partial}(\mathbf{B}, \mathbf{G}, r) &= \sum_{m \in \mathcal{N}} \ell_c(c_m, c'_m) \\ &+ \sum_{m \in \mathcal{P} \bigwedge \mathcal{O}(r)} \ell_c(c_m, c'_m) \\ &+ \lambda \sum_{m \in \mathcal{P} \bigwedge \mathcal{O}(r)} \|\mathbf{p}_m - \mathbf{p}'_m\|_2^2. \quad (2)\end{aligned}$$

The cost function for a frame is expressed as:

$$\ell(\mathbf{B}, \mathbf{G}) = \sum_{r=1}^{H} \ell_{partial}(\mathbf{B}, \mathbf{G}, r). \quad (3)$$

Here $\mathbf{G}$ is the corresponding ground-truth parsing results, i.e., $\mathbf{G} = (\mathbf{g}_1; \mathbf{g}_2; \cdots; \mathbf{g}_M)$ and $\mathbf{g}_m = (\mathbf{p}'_m; c'_m)$. $\mathcal{N}$ and $\mathcal{P}$ denote the un-matched (according to the invisible object entities in the ground-truth $\mathbf{G}$, namely $c'_m = 0$) and matched (according to the visible object entities in the ground-truth $\mathbf{G}$, namely $c'_m = 1$) subset of detection bounding boxes from $\mathbf{B}$. $\ell_c(\mathbf{y}, \mathbf{y}') = -\sum_i y_i \log(y'_i) - (1 - y_i)\log(1 - y'_i)$ is the cross-entropy cost. $\mathcal{O}(r)$ is the set of first r elements from the ordered matching list from the subset $\mathcal{P}$. To order the matching list, we define an ordering based on the bounding box overlapping between ground-truths and predictions (IoU). The last term is a displacement between the predicted bounding boxes' positions with the ground-truth ones. λ is a balance factor which is decided by cross-validation using a validation set. During network training, after each round of object bounding box prediction generation, we perform predicted bounding box matching and ordering with the ground-truth and proceed to further optimize the network model parameters. For testing, we take the last step prediction $\mathbf{B}_t^{(H)}$ for frame t to be the confirmed object parsing result. Bounding boxes with low confidence values are discarded.

The working mechanism of our network is well reflected by our designed cost function. For the first iteration ($l = r = 1$), the cost function only attempts to minimize the displacement between the top matched prediction bounding box $\mathcal{O}(1)$ to its corresponding ground-truth. The philosophy is that in the first step, we only need to detect the *easiest* object. When the iteration index increases, our cost function requires that more predicted bounding boxes should match their corresponding ground-truths. For every next iteration, we add one more object in the *must-match* list until all detections have been considered. Through this way, the challenging interactional object parsing problem is decomposed into a series of sub-problems with increasing difficulty level. As more and more object detections have been confirmed, we have more confidence to detect more *difficult* object based on its contextual relationship with previously detected objects. Once more object information gathered in the path, previous difficult object could also become *not that difficult* to detect. Note that for the predicted bounding boxes which do not have a match in the ground-truth, i.e., occluded part/object, our cost function can directly penalize this erroneous detections based on the first cross-entropy term (to minimize the confidence score associated with that object).

Discussions. Note that our formulation of progressively object detection scheme is different from the scheme proposed in the work [26]. In [26], only one type of object is considered. Therefore, after each round of prediction during model training, candidate object bounding boxes should be matched to the ground-truth bounding boxes through a bipartite graph matching algorithm. In contrast, for our problem, different types of objects are involved and we have a fixed object indexing scheme, i.e., each $\mathbf{b}_m$ in $\mathbf{B}$ is a fixed entity (hand, oil box, knife, etc). In other words, $\mathbf{b}_m$ could be only matched to its corresponding ground-truth $\mathbf{g}_m$, i.e., hand prediction matched to hand ground-truth. Moreover, the working mechanism of our network and the one proposed in [26] is different. The purpose of [26] is to sequentially output all object detections at some receptive field, one at each iteration; in contrast, our purpose is to gradually refine the entire object parsing vector $\mathbf{B}$. In each step, some bounding boxes (might be more than one) in $\mathbf{B}$ could be refined (refinement is performed on the entire vector $\mathbf{B}$).

Second, previous interactional object parsing method cannot well handle the scenario when some parts are occluded. For example, as revealed by the work [18], objects and body parts tracking easily gets failed when occlusion is serious. In contrast, since our model naturally handles the occlusion problem. Specifically, our method confirms object/part localizations from a sequential manner with a non-fixed order: simple objects could be consolidated first, difficult objects could be consolidated later, and those totally occluded ones could be explicitly decided as non-detection.

Model Training Details We use the open source package Caffe [9] (for the image feature learning part) and NLP-Caffe [1] (modified, for the LSTM part, similar to the usage in [26]) and train the proposed network using stochastic gradient descent (SGD) and back propagation through time (BPTT). The CNN feature (VGG-19) model is initialized by the ImageNet pre-trained model. The parameters of the LSTM part are randomly initialized in the range $[-1, 1]$. Both the CNN part and LSTM part are jointly trained. The temporal batch size is set as 32. We use an equal learning rate for all layers. The learning rate is initialized at 0.1 (the momentum is set as 0.8) and it is adjusted manually by dividing 10 when the validation error rate stops decreasing with the current learning rate. The network converges

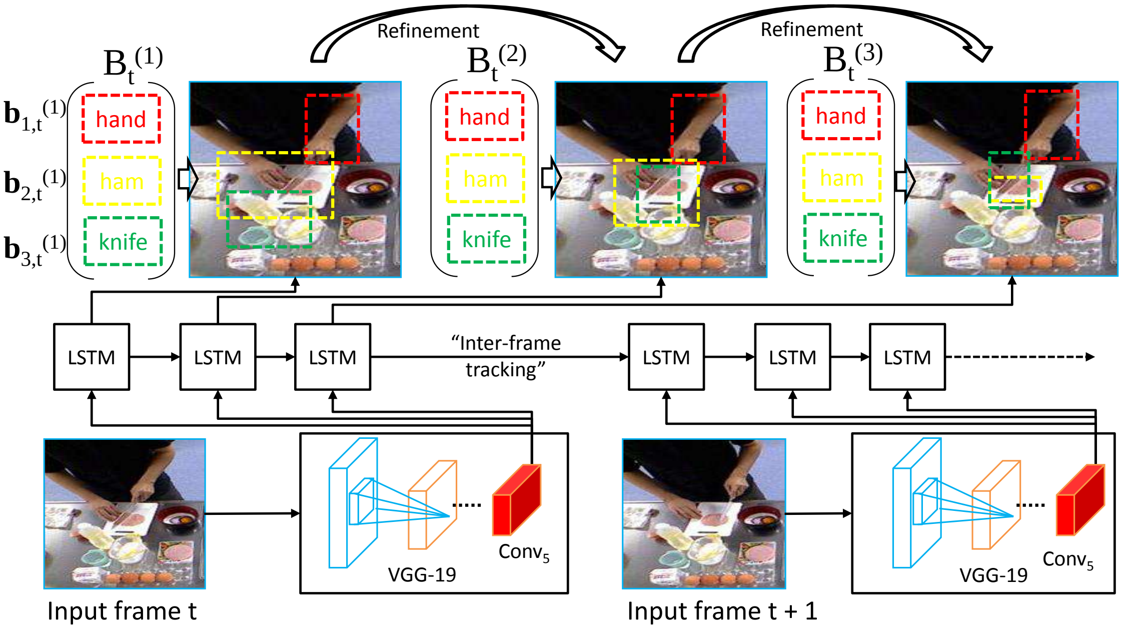

Figure 1. Overview of the proposed progressive interactional object parsing method based on LSTM network. We note that for the first iteration $\mathbf{B}_t^{(1)}$ (output from the first LSTM node), the inferred position for hand is accurate (since it is easy to detect hand); however, the other two objects (ham and knife) is NOT accurately localized (this is because of the occlusion, illumination variation and less-discriminative appearance of these objects). After several iterations ($\mathbf{B}_t^{(2)}$ and $\mathbf{B}_t^{(3)}$) the refined locations of these objects are getting more accurate, thanks to the proposed progressively refinement scheme. In this example, $H = M = 3$.

roughly with 30 epochs, and each epoch contains 10000 iterations. It takes around 10 days on one NVIDIA Tesla $K40$ GPU. During testing, parsing each frame requires around 2 seconds.

4. Fine-grained Action Detection

We develop a fine-grained action detection method based on the interactional parsing results. A video is divided uniformly by overlapped segments of length 30, 60, 90 frames (detection windows). Temporal overlappings are of 10, 20 and 30 frames. Within each video, we pool the motion features within each parsed body part/object to achieve part/object specific motion representation, similar to the scheme used in [18]. It has been widely proved in previous works that this type of object-specific motion feature pooling method can significantly outperform global motion pooling method. This is because object/body part centric motion pooling is less ambiguous than global pooling method.

The local motion features we use are the improved dense motion trajectories [31]. For each trajectory, we extract histogram of oriented gradient (HoG), motion boundary histogram (MBH), histogram of optical flow (HoF) and trajectory shape (TS) descriptors as in [30]. We perform PCA to reduce the dimension of each descriptor by half. These features are encoded using improved Fisher vector with the number of clusters $K = 128$. We also apply a second PCA to reduce the overall Fisher vector encoding by a factor of 0.1 (i.e., keep around 90% energy). Assume that the dimensionality of Fisher vector is d, number of objects of interest is M, then a video segment i is represented by a $d \times M$ dimensional vector $\mathbf{x}_i$. We then use linear SVM learned on this segment level representation to detect the action label of every video segment. The inferred labels are averaged over three detection window scales.

5. Experiments

Our experiments are focused on two parts. On the one hand, we compare our fine-grained interactional object parsing result to that of the state-of-the-art object tracking algorithms. On the other hand, we show the fine-grained action detection performance based on our proposed framework. Similar to the work [18], all experiments are performed on two challenging interaction-intensive fine-grained action benchmarks as follows.

1) **ICPR 2012 Kitchen Scene Context based Gesture Recognition dataset (KSCGR) [2]**. There are five candidate cooking menus cooked by five different actors. Each of the videos are from 5 to 10 minutes long containing $9,000$ to $18,000$ frames. The dataset contains eight types of cooking motions such as *baking*, *boiling*, *breaking*, etc. Following [18], the objects of interest (which we parse) are *fry pan*, *oil bottle*, *salt bottle*, *bowl*, *knife*, *spoon*, *chopstick*, *spatula*, *chopping board*, *egg* and *ham*. For KSCGR dataset, the e-

valuation metric of action detection is the mean recognition F-score over all action categories.

2) **MPII Fine-grained Kitchen Activity Dataset (MPI-I) [22]**. It contains 65 different cooking activities, such as *cut slices*, *pour spice*, etc., recorded from 12 participants. In total there are 44 videos with a total length of more than 8 hours or $881,755$ frames. The dataset contains a total of $5,609$ annotations of 65 activity categories. Following [18], the objects that we parse are *bottle*, *bowl*, *bread*, *charger*, *electric range*, *cup*, *cupboard*, *chopping board*, *dough*, *drawer*, *egg*, *lid*, *food wrapper*, *knife*, *pan*, *slicer*, *plate*, *pot*, *blender*, *seasoning bottle*, *bottle rack*, *juicers*, *tin*, *tin opener* and *towel*. For MPII dataset, we follow experimental configuration and evaluation metric defined by the dataset developer [22]. In brief, leave-one-person-out cross validation is used.

For object/body part annotations, we obtained from the authors of [18] around 20000 frames of annotations from the training data. These annotations are used to train all comparing detectors/trackers.

5.1. Results on Interactional Object Parsing

We use the same annotated testing sequences as in [18] on the testing set of KSCGR to evaluate the interactional object parsing performance. These sequences are manually annotated with object positions (bounding boxes). According to [18], these selected testing sequences are representative sequences which contain all types of human object interactions with frequent occlusions. It is a good test bed for evaluating our interactional object parsing algorithm as well as other object tracking algorithms.

The proposed interactional object parsing method is compared with the state-of-the-art tracker specifically designed for fine-grained action detection (IAT) [18] (which jointly tracks hand and objects). According to [18], we use the same measuring metrics as follows:

1. Average Distance Error (Err.): the average distance between the center of the identified bounding box and the center of the ground-truth bounding box;

2. Precision (Prec.): the average percentage of frames for which the overlap between the identified bounding box and the ground-truth bounding box is at least 50 percent.

Table 1 compares the measurements averaged over all target objects and over all frames in the video sequence. Figure 2 shows several examples of the tracked/parsed results using IAT and our algorithm, for qualitative comparison. To reveal the working mechanism of our progressive object parsing refinement algorithm, Figure 3 plots the change of object localization error with respect to the number of LSTM iteration steps. In the meantime, in Figure 4 we also show several examples of the *refinement* process of our interactional object parsing algorithm.

Sequence	IAT		Ours	
	Err.	Prec.	Err.	Prec.
baking (3786)	28.9	0.56	24.5	0.60
boiling (3320)	25.5	0.59	25.0	0.61
breaking (299)	20.4	0.64	17.3	0.72
cutting (1373)	24.8	0.66	20.1	0.70
mixing (705)	17.9	0.68	17.4	0.69
peeling (3241)	30.1	0.62	28.9	0.64
seasoning (303)	12.3	0.69	13.9	0.66
turning (3402)	15.4	0.71	15.0	0.73

Table 1. Comparisons of the object localization performances of various object parsing or tracking methods. Numbers of frames are indicated in brackets.

We make several key observations from the results in Table 1 and Figure 2, 3, 4. First, our interactional object parsing method outperforms prior art (although the IAT tracker also performs quite well in most of the cases). This is because that progressively refining the interactional object detection results can break the difficult parsing task down to easier steps. Quantitatively, Figure 3 shows that the average object localization (parsing) error decreases when the refinement process goes *deeper*. We also note from Figure 4 that easy object such as hand is first localized and then the hand position provides rich contextual information to localize other interacting objects such bowl, knife and chopsticks, i.e., following an easy-to-difficult manner. Second, we note that for some action sequences such as *cutting*, our method outperforms the IAT significantly (see Table 1). This is because that it is usually not easy to localize the knife when it is partly occluded by the operating hand, i.e., the knife's color feature often confuses with the background. In contrast, our algorithm first localizes the hand and then uses the contextual information to localize the knife, therefore the parsing accuracy is higher. This could be also observed from the 8-th and 10-th examples in Figure 2. We see that for the IAT tracker, when the object knife and chopstick are occluded, it easily gets lost during tracking. In contrast, since our algorithm well models the contextual information between hand and the operated objects, this issue could be alleviated. Third, in some cases when there exist ambiguous objects, both the IAT algorithm and our proposed algorithm could fail. For example, our algorithm also mistakes fry pan handler as spoon in the last example of Figure 2. This is because the appearances of both objects are quite similar and their spatial relationships with respect to hand are also similar during certain action.

We also compare our algorithm with the R-CNN and fast R-CNN algorithms in our off-line experiment. Results show that R-CNN and fast R-CNN perform worse than our proposed method, provided that all the experimental settings

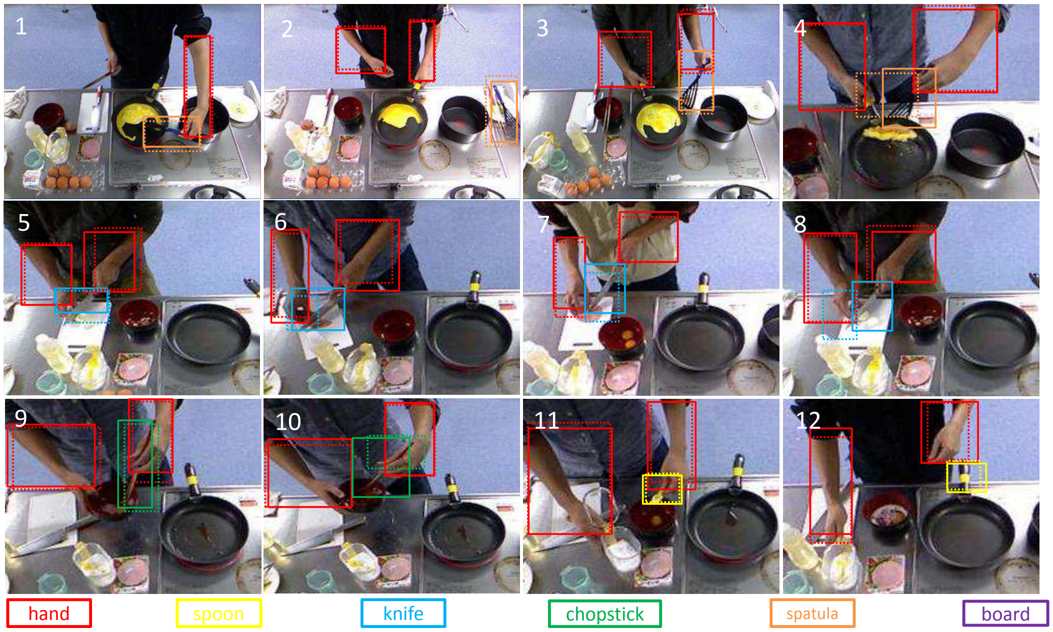

Figure 2. Twelve examples of the interactional object parsing results of our method (solid line bounding boxes) and the IAT method (dashed line bounding boxes).

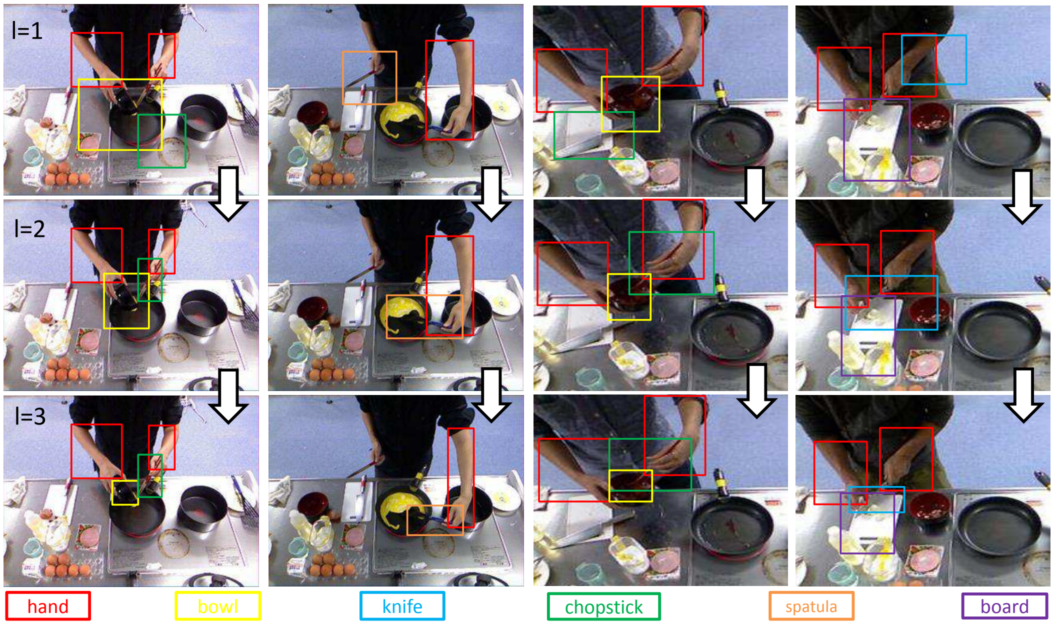

Figure 4. Four examples of the progressive interactional object parsing refinement process of our algorithm. Each column corresponds to a frame for parsing. From top to down, it shows the refinement process, i.e., $l = 1, 2, 3$. Note that hand is always easily localized first and other objects which have interaction with hands will obtain more and more precise localization after several LSTM iterations.

are the same. This is due to two reasons: 1) R-CNN based methods do not consider contextual information between objects; 2) Some objects in the fine-grained video are too small or of strange aspect ratio so that R-CNN based methods can not well handle it; in contrast, our proposed method uses a progressive easy-to-difficult detection scheme so that it can deal with these difficult cases. Also, our off-line experiment shows that jointly applying LSTM on consecutive frames (tracking) performs better than simply applying LSTM on individual frames.

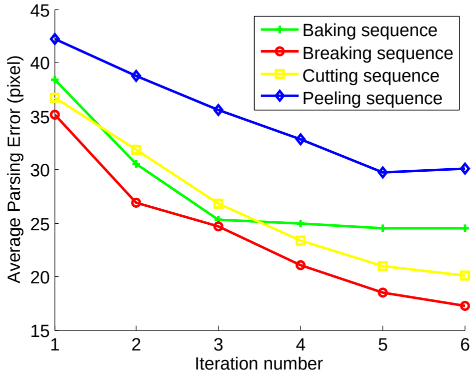

Figure 3. Illustration of the interactional object parsing error decreasing procedure using our iterative refinement method.

Method	Doman and Kuai [2]	IDT-IFV-SVM	IAT-Action [18]	Ours
Mean F-score	0.74	0.76	0.79	**0.84**

Table 2. Detection performance (mean F-score for all classes) comparisons for KSCGR dataset.

Approach	Prec.	Recall	AP
Rohrbach et al. [22]	19.8	40.2	45.0
IDT-IFV-SVM	24.5	46.8	50.7
IAT-Action [18]	28.6	48.2	54.3
Ours	**34.8**	**51.7**	**58.9**

Table 3. Detection performance comparisons for MPII dataset.

5.2. Results on Fine-grained Action Detection

We apply the object-centric motion feature pooling introduced in Section 4 for action detection on the two benchmark fine-grained action datasets. We compare our algorithm with the state-of-the-art algorithms on fine-grained action detection. The comparing algorithms include: 1) the baseline globally pooled Fisher vector representation with linear SVM detector (the parameters of motion feature descriptors, Fisher vector construction [21] and temporal sliding window are exactly the same as our proposed algorithm, denoted as IDT-IFV-SVM); and 2) the multiple-granularity based fine-grained action detection algorithm (denoted as IAT-Action) [18]. On the KSCGR dataset, we also compare our method to the best reported result in the contest by Doman and Kuai [2]. Comparison results are shown in Table 2. Detection mean F-scores are reported on the KSCGR dataset. On the MPII dataset, we also compare our method to the best reported result in [22] by Rohrbach et al. Multiclass precision (Pr) and recall (Rc). The mean value of single class average precision (AP) are reported in Table 3.

We note that using the object-centric motion pooling method based action detection framework (including ours and IAT-Action) boosts the detection performances, compared with using the traditional global pooling scheme. This demonstrates that the object and body part parsing algorithm is precise enough to achieve high performance object-centric motion feature pooling and representation. Moreover, our proposed fine-grained action detection framework also outperforms the state-of-the-art IAT-Action method on both benchmarks. This is because our algorithm performs better object parsing and the pooled action representation is therefore more discriminative.

6. Conclusions

In this work, we proposed a novel progressive interactional object parsing method based on the recurrent neural network (LSTM), for the application of fine-grained action analysis. Experiments on two benchmark datasets demonstrated good parsing accuracy of our algorithm compared with prior art. Based on the good parsing results, we also achieved the state-of-the-art fine-grained action detection performances on those benchmarks.

Acknowledgement

This work was Supported by National Natural Science Foundation of China (Grant No.: 61502301), and China's Thousand Youth Talents Plan for Dr. Bingbing Ni. It was also partly supported by the Shanghai Pujiang Talent Program (No.15PJ1405700), and NSFC (No. 61502304).

References

[1] http://github.com/russell91/nlpcaffe.

[2] http://www.murase.m.is.nagoya-u.ac.jp/kscgr/index.html.

[3] M. de La Gorce, N. Paragios, and D. Fleet. Model-based hand tracking with texture, shading and self-occlusions. In *CVPR*, 2008.

[4] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, and K. S. T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. 2014.

[5] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *T-PAMI*, 32(9):1627–1645, 2010.

[6] A. Graves, A.-R. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *ICASSP*, pages 6645–6649, 2013.

[7] A. Gupta and L. Davis. Objects in action: An approach for combining action understanding and object perception. In *CVPR*, 2007.

[8] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[9] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv:1408.5093*, 2014.

[10] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *arXiv:1412.2306*, 2014.

[11] H. Kjellström, J. Romero, D. Martínez, and D. Kragić. Simultaneous visual recognition of manipulation actions and manipulated objects. In *ECCV*, pages 336–349, 2008.

[12] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from rgb-d videos. *CoRR*, 2012.

[13] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, 2003.

[14] J. Lei, X. Ren, and D. Fox. Fine-grained kitchen activity recognition using rgb-d. In *ACM Conference on Ubiquitous Computing*, pages 208–211, 2012.

[15] C. Li and K. M. Kitani. Pixel-level hand detection in ego-centric videos. In *CVPR*, 2013.

[16] X. Liang, L. Lin, and L. Cao. Learning latent spatio-temporal compositional model for human action recognition. In *ACM Multimedia*, 2013.

[17] D. Moore, I. Essa, and M. Hayes. Exploiting human actions and object context for recognition tasks. In *ICCV*, Corfu, Greece, 1999.

[18] B. Ni, V. R. Paramathayalan, and P. Moulin. Multiple granularity analysis for fine-grained action detection. In *CVPR*, pages 756–763, 2014.

[19] J. C. Niebles, H. Wang, and L. Fei-fei. Unsupervised learning of human action categories using spatial-temporal words. In *BMVC*, 2006.

[20] B. Packer and D. Koller. A combined pose, object, and feature model for action understanding. In *CVPR*, 2012.

[21] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, pages 143–156, 2010.

[22] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *CVPR*, pages 1194–1201, 2012.

[23] M. S. Ryoo and J. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *ICCV*, pages 1593–1600, 2009.

[24] J. Schmidhuber, F. Gers, D. Eck, J. Schmidhuber, and F. Gers. Learning nonregular languages: A comparison of simple recurrent networks and lstm. *Neural Computation*.

[25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[26] R. Stewart and M. Andriluka. End-to-end people detection in crowded scenes. *CoRR*, abs/1506.04878, 2015.

[27] H. Trinh, Q. Fan, P. Gabbur, and S. Pankanti. Hand tracking by binary quadratic programming and its application to retail activity recognition. In *CVPR*, pages 1902–1909, 2012.

[28] V. Veeriah, N. Zhuang, and G. Qi. Differential recurrent neural networks for action recognition. *CoRR*, abs/1504.06678, 2015.

[29] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. *arXiv:1411.4555*, 2014.

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

[31] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[32] Y. Wang and G. Mori. Hidden part models for human action recognition: Probabilistic versus max margin. *T-PAMI*, 33(7):1310–1323, 2011.

[33] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. Rehg. A scalable approach to activity recognition based on object use. In *ICCV*, 2007.

[34] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye, and X. Xue. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. *Proceedings of ACM Multimedia*, 2015.

[35] J. Yuan, Z. Liu, and Y. Wu. Discriminative video pattern search for efficient action detection. *T-PAMI*, 33(9):1728–1743, 2011.

[36] 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.

[37] Y. Zhou, B. Ni, S. Yan, P. Moulin, and Q. Tian. Pipelining localized semantic features for fine-grained action recognition. In *ECCV*, pages 481–496, 2014.

[38] S.-C. Zhu and D. Mumford. A stochastic grammar of images. *Foundations and Trends in Computer Graphics and Vision*, 2(4):259–362, 2006.