Interaction Part Mining: A Mid-Level Approach for Fine-Grained Action Recognition

Yang Zhou[1], Bingbing Ni[2], Richang Hong[3], Meng Wang[3], and Qi Tian[1]
[1]University of Texas at San Antonio, US
[2]Advanced Digital Sciences Center, Singapore
[3]HeFei University of Technology, China
myh511@my.utsa.edu bingbing.ni@adsc.com.sg hongrc@hfut.edu.cn
eric.mengwang@gmail.com qi.tian@utsa.edu

Abstract

Modeling human-object interactions and manipulating motions lies in the heart of fine-grained action recognition. Previous methods heavily rely on explicit detection of the object being interacted, which requires intensive human labour on object annotation. To bypass this constraint and achieve better classification performance, in this work, we propose a novel fine-grained action recognition pipeline by interaction part proposal and discriminative mid-level part mining. Firstly, we generate a large number of candidate object regions using off-the-shelf object proposal tool, e.g., BING. Secondly, these object regions are matched and tracked across frames to form a large spatio-temporal graph based on the appearance matching and the dense motion trajectories through them. We then propose an efficient approximate graph segmentation algorithm to partition and filter the graph into consistent local dense sub-graphs. These sub-graphs, which are spatio-temporal sub-volumes, represent our candidate interaction parts. *Finally, we mine discriminative mid-level part detectors from the features computed over the candidate interaction parts. Bag-of-detection scores based on a novel* Max-N pooling *scheme are computed as the action representation for a video sample. We conduct extensive experiments on human-object interaction datasets including MPII Cooking and MSR Daily Activity 3D. The experimental results demonstrate that the proposed framework achieves consistent improvements over the state-of-the-art action recognition accuracies on the benchmarks, without using any object annotation.*

1. Introduction

In recent years, fine-grained action recognition has raised extensive research [21, 13] due to its potential applications in assisted daily living, medical surveillance and smart home.

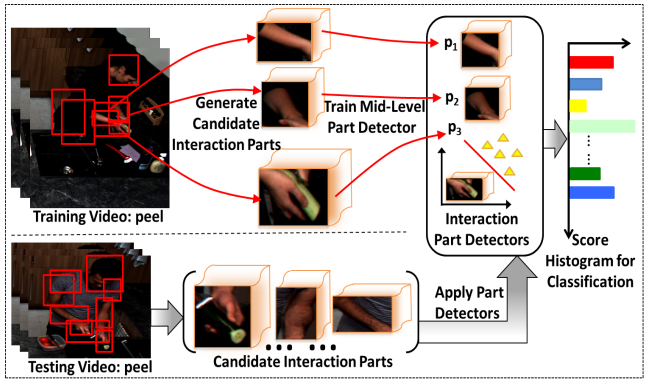


Figure 1: Our mid-level approach for fine-grained action recognition includes interaction part formation to generate candidate human-object interaction parts, and interaction part mining to train discriminative exemplar part detectors.

On one hand, fine-grained manipulation actions involve a large amount of interactions between human and objects. Therefore, how to model the interactions between human and objects plays a critical role in action representation and recognition. Many research works [36, 16, 17, 33, 32, 11, 35, 7, 10, 20, 21, 4, 24] have devoted to modeling the contextual information between human and objects/scenes for action recognition. However, to model human and object contextual information, explicit detection of objects is often required by the above methods. Training these object detectors requires labour-extensive human annotation work. In fine-grained action recognition where many types of objects are manipulated in a single action, it is not feasible to 1) label the training object instances; and 2) detect those objects with decent detection accuracy.

On the other hand, the spatio-temporal features [12], especially the recent proposed dense trajectories [29] encoded

with naive BoW [6] or Fisher vector representation [19] are commonly used for action recognition, but they can only capture the characteristics of the global motion in the entire video volume. The low-level feature extraction with global pooling might not be suitable for representing fine-grained actions because this global presentation easily attenuates the important localized interaction motion within the background movement. In contrast, for fine-grained motion, it is more important to highlight what kind of interaction motion is being performed in a local spatio-temporal sub-volume of the video. Therefore, we need a mid-level representation to capture the important local human-object interaction motion.

To address the above two issues, we propose a novel mid-level based pipeline for fine-grained action representation and classification, which is motivated by two recent successes in visual recognition, *i.e*., object proposal technique [2] and mid-level discriminative visual element mining [9, 8, 3]. We show the general framework in Figure 1.

Firstly, we utilize the off-the-shelf object proposal BING [2] to generate a large number of object proposals (*i.e*., candidate regions) on the segmented foreground motion pixels. Then we construct a spatio-temporal graph by matching the object proposal spatio-temproally based on both dense trajectory linkage and appearance similarity. An efficient graph segmentation algorithm is proposed to group the object regions which are temporally continuous and spatially compact into spatio-temporal sub-volumes. We name these sub-volumes as *interaction parts*. The extracted interaction parts have two advantages: 1) the extraction procedure is unsupervised which means object annotation is not required; and 2) the extracted interaction parts naturally contain mid-level information on what object and what motion is being performed.

Secondly, for each interaction part, we compute a Fisher vector representation by pooling the spatio-temporal motion features of the sub-volume. We learn a set of discriminative part detectors using an improved version of the image block mining approach [9], *i.e*., to seed discriminative interaction parts by considering both appearance and motion features. Finally, we utilize the trained part detectors to score the interaction parts within each video, and summarize the part scores for each video sample using a generalization of max pooling technique, *i.e*., Max-N pooling. We validate our mid-level based video representation on the MPII Cooking [26] and MSR Daily Activity 3D [30] datasets. The results show that our mid-level approach outperforms the state-of-the-art performance on both datasets. It is more effective than low-level features in capturing human-object interaction motion, and it is even better than the previous approach [38] which requires object detection.

We conclude our contribution as follows. We propose a mid-level video representation for fine-grained action recognition. The method effectively captures the significant human-object interaction motion by object proposal based interaction part formation and discriminative interaction part mining. Most importantly, our method is free of explicit object detection, which gives us three major advantages: 1) it is more stable than the object detection approaches which are quite affected by the detection accuracy to different objects; 2) it saves extensive human labour for object annotation; and 3) it is quite applicable and feasible in real problems. Furthermore, we propose a novel Max-N pooling which improves performance compared to the previous naive max pooling approach.

The rest of this paper is organized as follows. We list some closely related works in Section 2. The details of our mid-level interaction part mining pipeline are described in Section 3. We demonstrate the experimental setting and results in Section 4. Conclusion is given in Section 5.

2. Related Work

There exists several pioneer works on using mid-level representation for action recognition. Michalis *et al*. [25] apply spatial grouping to trajectories to form action parts, but the grouping is not semantic to capture human-object interactions. Motion atoms and phrases [31] coarsely uses temporally segmented video sub-volumes as their action proposals which are not able to highlight the human-object interaction motion. Wang *et al*. [30] model human action by human joint features, but the approach needs skeleton and depth information from sensors and trackers. Some other action part mining approaches (*e.g.,* [4, 27, 34]) mine sparse action part using latent structural models, which in practice is not flexible for real problem. On the contrary, a dense representation like our proposed approach is much more discriminative. Recently, Zhou *et al*. [38] propose the semantic trajectories, a good step in fine-grained action recognition, however, this method requires extensive human labour to annotate the dataset for object detection.

Secondly, recent image mid-level patch mining approaches [9, 28, 3, 15] show success in image recognition tasks. However, these mid-level mining techniques have not been applied to learn discriminative spatio-temporal interaction motion volumes in videos. Peng *et al*. [22] use densely sampled spatio-temporal video sub-volumes as action proposals, however it is not applicable to the fine-grained interaction recognition problem because their sampling method ignores the object information, it is rather a random sampling method. Gupta *et al*. [8] propose a discriminative patch mining method for action recognition, but it is still image based, *i.e*., image patch mining. In contrast, our approach is to directly find spatio-temporal volumes that are rich in human and object motion, instead of linking image patches into sequence.

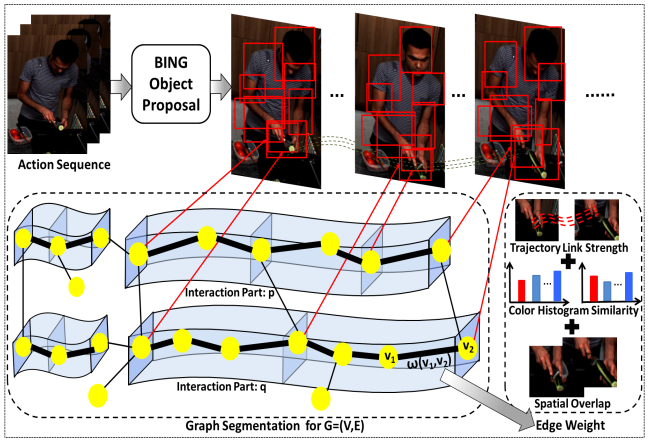


Figure 2: Procedure for interaction part formation. We generate large amounts of object proposals on foreground motion of each frame, the object regions are tracked by dense trajectories and appearance matching to form graph $G = (V, E)$. Thicker edge indicates larger similarity between two nodes. The graph is segmented into interaction parts by grouping object regions that share similar appearance, spatial compactness and strong trajectory link.

3. Methodology

Our idea is to learn mid-level interaction motion representation for fine-grained action video. The key observation is that interesting interaction occurs within the spatio-temporal volume surrounding the object being manipulated. Therefore, the key step is to *propose* candidate spatio-temporal sub-volumes of interaction (named as interaction parts), then apply mid-level mining approach to capture discriminative interaction part detection models. To bypass the intensive human labeling work for object detection, we develop a novel interaction part formation pipeline. Firstly, we utilize the off-the-shelf object proposal generation tool to generate object region candidates for each frame. Then we utilize dense trajectories to temporally link these object proposals and form the spatio-temporal region graph. Finally, we develop a simple yet effective graph segmentation algorithm to obtain strongly connected sub-graphs. These sub-graphs correspond to moving object regions that are appearance consistent and temporally continuous, *i.e.*, the target spatio-temporal sub-volumes representing the candidate interactions. The details of our method are given as follows.

3.1. Interaction Part Formation

We show the procedure of interaction part formation in Figure 2. To get rid of background motion and filter out still objects (which are not being interacted), we apply foreground motion region segmentation using the state-of-the-art method based on Gaussian mixture model background subtraction [39] as a pre-processing step.

We utilize the off-the-shelf object proposal tool BING [2] to generate a large number of object proposals (*i.e.*, candidate regions) on the segmented foreground motion regions. We follow the same parameter setting of BING [2] to detect the object regions for each video frame. The generated object proposals which have sufficient overlapping ($> 50\%$) with the detected foreground motion regions are selected. We keep 5000 object proposals for each video frame. We also remove some bounding boxes which are too large (*i.e.*, larger than 100×100), or in skewed size ratio (*i.e.*, larger than 6:1).

Next step, we match and track these object regions across frames to build a large spatio-temporal graph based on the appearance matching and the dense motion trajectories through them. For each video sample, we build a graph denoted as $G = (V, E)$. $V = \{v_1, v_2, \ldots\}$ are the vertex nodes and they are constructed from the object regions extracted across the video frames. $E = \{e_1, e_2, \ldots\}$ are the edges. The edge weight $\omega(v_1, v_2)$ between two nodes v_1 and v_2 are computed by matching their appearances and the linkage between them through dense trajectories. Namely, the edge weight is the linear combination of: 1) spatial overlap $s(v_1, v_2)$; 2) trajectory link strength $l(v_1, v_2)$; and 3) appearance similarity $a(v_1, v_2)$, which is formulated as:

$$\omega(v_1, v_2) = \begin{cases} \lambda_1 \cdot s(v_1, v_2) + \lambda_2 \cdot l(v_1, v_2) + \lambda_3 \cdot a(v_1, v_2), \\ \qquad\qquad |t(v_1) - t(v_2)| \leq 1 \\ -\infty, \qquad\qquad otherwise \end{cases} \tag{1}$$

where $t(v)$ is the time frame of node v. λ_1, λ_2, and λ_3 are the weights. We set these weights by cross-validation on the training dataset. Although matching more than two frames is also an option, in this work, we only connect nodes that are in the same frame or two consecutive frames, *i.e.* when $|t(v_1) - t(v_2)| \leq 1$. We set $-\infty$ weight to edge $e(v_1, v_2)$ if v_1 and v_2 are neither in the same frame nor two consecutive frames. We explain these edge weight terms in detail as follows:

Spatial Overlap: $s(v_1, v_2)$ is computed as:

$$s(v_1, v_2) = \frac{r(v_1) \cap r(v_2)}{r(v_1) \cup r(v_2)}, |t(v_1) - t(v_2)| \leq 1, \tag{2}$$

where $r(v)$ is the area of object region v, then $s(v_1, v_2)$ is the Jaccard similarity of two region areas.

Trajectory Link Strength: The key observation is that if two object regions on two frames are linked by dense trajectories, they are highly likely belonging to the same object. $l(v_1, v_2)$ is computed as:

$$l(v_1, v_2) = \frac{2 \cdot traj(v_1, v_2)}{r(v_1) + r(v_2)}, |t(v_1) - t(v_2)| \leq 1, \tag{3}$$

where $traj(v_1, v_2)$ is the number of trajectories to traverse object region v_1 and object region v_2, the denominator is the summation of region areas $r(v_1)$ and $r(v_2)$. Note that the

trajectory link is stronger when denser trajectories traverse through v_1 and v_2.

Appearance Similarity: To calculate $a(v_1, v_2)$, we compute the HSV color histograms of v_1 and v_2, apply l_2 normalization to them and get $H(v_1)$ and $H(v_2)$. Then we compute their correlation coefficient as:

$$a(v_1, v_2) = \frac{\text{cov}(H(v_1), H(v_2))}{\sigma(H(v_1)) \cdot \sigma(H(v_2))}, |t(v_1) - t(v_2)| \leq 1, \tag{4}$$

where $\text{cov}(H(v_1), H(v_2))$ is the covariance of $H(v_1)$ and $H(v_2)$, $\sigma(H(v_1))$ and $\sigma(H(v_2))$ are the standard deviations of $H(v_1)$ and $H(v_2)$ respectively.

Algorithm 1 Approximate Graph Segmentation Algorithm for Interaction Part Formation.

Input: $G = (V, E)$

```
for v ∈ V do
  V' ⇐ {v'|v' ∈ V − v}
  for v' ∈ V' do
    if ω(v, v') ≤ τ ∧ t(v1) = t(v2) then
      E ⇐ E − e(v, v')
    else if ω(v, v') ≤ ε ∧ |t(v1) − t(v2)| = 1 then
      E ⇐ E − e(v, v')
    end if
  end for
end for
Connected ⇐ FindConnectedComponents(G)
I ⇐ DensifyInteractionParts(Connected)
```

Output: I $(Interaction\ Parts)$

With the above graph G, the task of generating candidate interaction parts is equivalent to partitioning and filtering the graph into consistent local dense sub-graphs, since such sub-graphs contain moving objects or body parts which have consistent appearances and can be tracked over frames. To this end, we propose an efficient approximate graph segmentation algorithm. For each interaction part, we require the object regions to be temporally continuous, spatially compact, strongly linked and share similar appearances. This corresponds to large edge weights in the graph. We require two nodes of the same interaction part to share very large weight ($\omega(v_1, v_2) > \tau$) if they are in the same frame ($t(v_1) = t(v_2)$). We set $\tau = 0.75$ to enforce nodes of the same frame to be extremely compact. We also require two nodes of consecutive frames ($|t(v_1) - t(v_2)| = 1$) to share weight as $\omega(v_1, v_2) > \varepsilon$. We set $\varepsilon = 0.25$ to ensure compactness, and to allow motion displacement between two consecutive frames. Finding an exact solution as [1] is computationally expensive. Considering that the action video database is often large-scale, we propose an approximate algorithm to segment the graph G to interaction parts in terms of our criteria. We show the graph segmentation algorithm in Algorithm 1. Firstly, we remove the edges not meeting our criteria. Secondly, we find the connected components as the segmented graphs. In the DensifyInteractionParts step, we densely sample sub-graphs from the segmentations. We cut a large sub-graph (*i.e.*, long frame range) to small ones by different length scale (15, 20, 25, 30, 35, 40, 45, 50) respectively, with 75% overlap between neighbors for each scale. The dense sub-graphs including the original segmentations are utilized to form the interaction parts I.

To represent each candidate interaction part (*i.e.*, to form mid-level feature descriptor), we adopt the method of Fisher vector coding of dense trajectories [29]. We densely extract trajectories within each candidate interaction part using the default parameters as in [29]. Motion features, *i.e.*, MBH (96-dim for MBHx and 96-dim for MBHy) and HOG (96-dim) are computed from the extracted dense trajectories. We apply PCA to reduce feature dimension from 96 to 64. We train Gaussian mixture models with 64 components, and encode the interaction parts with improved Fisher vectors [19]. Following [23], we also apply square-rooting and normalization for the computed Fisher vectors. Finally, each candidate interaction part is described by a 8192-dimensional (*i.e.*, $64 \times 64 \times 2$) vector $\mathbf{x}$. A video sample v contains a bag of mid-level features $\mathbf{X}_v = \{\mathbf{x}_1, \mathbf{x}_2, \ldots\}$.

3.2. Interaction Part Mining

To mine discriminative mid-level interaction part detectors from the set of candidate parts extracted from the training video dataset, we leverage the strategy proposed in [9], that includes seeding, expansion, and selection. In particular, the previous heuristic seeding scheme in [9] seeds large number of image patches by only considering the appearance (*i.e.*, HOG), which is not suitable for the spatio-temporal parts in our problem. Also, training for all parts is computationally expensive and unnecessary. Therefore, we propose a new seeding scheme based on *tf-idf* score to only seed the discriminative parts considering both appearance (*i.e.*, HOG) and motion (*i.e.*, MBH) features. Details of seeding are given as follows.

Initially, we have the interaction part representations $\mathbf{X}_v = \{\mathbf{x}_1, \mathbf{x}_2, \ldots\}$ of the video sample v. First of all, we compute the nearest neighbors for each interaction part. Given two part representations, $\mathbf{x}_i = \{\mathbf{h}_i^c\}_{c=1}^3$ and $\mathbf{x}_j = \{\mathbf{h}_j^c\}_{c=1}^3$, c represents the channel of HOG, MBHx or MBHy, $\mathbf{h}^c$ is a D dimensional Fisher vector for c-th channel. The distance metric $K(\mathbf{x}_i, \mathbf{x}_j)$ between two interaction parts $\mathbf{x_i}$ and $\mathbf{x_j}$ is given as the weighted normalized Euclidean distances of three channels:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \sum_c w^c \cdot \exp\left(-\frac{d\left(\mathbf{h}_i^c, \mathbf{h}_j^c\right)}{A^c}\right), \tag{5}$$

where A^c is used for normalization of the channel, which is the mean value of the Euclidean distances between the

training samples for the c-th channel. The weight w^c is an empirical value. It can be simply set as the $1/c$ or give more weights to features that typically achieve better performance, *i.e.*, the MBH feature descriptors by empirical experience. $d\left(\mathbf{h}_i^c, \mathbf{h}_j^c\right)$ is the Euclidean distance between $\mathbf{h}_i^c$ and $\mathbf{h}_j^c$.

We rank the interaction parts by their discriminability. To require one interaction part $\mathbf{x}(\mathbf{x} \in \mathbf{X}_v)$ of action class $y(\mathbf{x})$ to be discriminative, the same label $y(\mathbf{x})$ (compared to other action classes) should account for the major part among its top retrieved nearest neighbors. Therefore, we compute the *tf-idf* score from the action labels of its top k nearest neighbors $knn(\mathbf{x})$. In our implementation, we set $k = 20$, the *tf-idf* score is $\frac{\left|\{\mathbf{x}'|\mathbf{x}' \in knn(\mathbf{x}) \wedge y(\mathbf{x}')=y(\mathbf{x})\}\right|}{|\{\mathbf{x}'|\mathbf{x}' \in knn(\mathbf{x})|}$, which is the ratio of interaction parts belonging to $y(\mathbf{x})$ among the top k nearest neighbors. Then we seed the interaction parts that rank higher with *tf-idf* scores.

After seeding some distinct interaction parts as exemplars, we then apply the same strategy as [9] to learn and select the discriminative exemplar detectors as our interaction part detectors. Each part detector $\mathbf{p}$ is the trained weight vector. M discriminative interaction part detectors are selected for each action class. Assuming that we have Q action classes, there will be $T = Q \times M$ part detectors in total. The part detectors $\mathbf{P}=\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_T\}$ are utilized to form the video representation in next step.

3.3. Max-N pooling

Given the learned discriminative interaction part detectors $\mathbf{P}=\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_T\}$, a popular way to form video level representation is to apply all the learned detectors onto all candidate interaction parts extracted within the video, *i.e.*, to score each candidate interaction part. Then for each part detector we maximumly pool its responses and concatenate all detector scores into a T-dimensional vector. Mathematically, given the interaction parts $\mathbf{X}_v=\{\mathbf{x}_1, \mathbf{x}_2, \ldots\}$ of the video sample v, the score $f(\mathbf{p}, v)$, *i.e.*, part detector $\mathbf{p}$ with respect to video v is given by max pooling the part scores:

$$score(\mathbf{x}, \mathbf{p}) = \langle \mathbf{x}, \mathbf{p} \rangle, \mathbf{x} \in \mathbf{X}_v, \mathbf{p} \in \mathbf{P}, \quad (6)$$

$$f(\mathbf{p}, v) = \max_{\mathbf{x} \in \mathbf{X}_v} score(\mathbf{x}, \mathbf{p}), \mathbf{p} \in \mathbf{P}, \quad (7)$$

where $\langle \mathbf{x}, \mathbf{p} \rangle$ is the inner product between part detector $\mathbf{p}$ and interaction part $\mathbf{x}$. The final video representation is $\mathbf{f}_v=[f(\mathbf{p}_1, v), f(\mathbf{p}_2, v), \ldots, f(\mathbf{p}_T, v)]$, which is an activation vector, scored by each part detector of $\mathbf{P}$. The activation vector $\mathbf{f}_v$ is used to represent the video sample v.

However, using max pooling sometimes might miss important responses from some candidate parts of the video. These parts might not give the max response values for a certain detector, however, they are still representative for a certain action class, and retaining the values in the final representation still adds discriminative information. Therefore, in this work, we propose a generalized approach for max pooling, called Max-N pooling. Namely, for each part detector $\mathbf{p}$, we keep its top N detection scores to the interaction parts. The $f(\mathbf{p}, v)$ is computed as:

$$f(\mathbf{p}, v) = \underset{\mathbf{x} \in \mathbf{X}_v}{\text{MaxN}} score(\mathbf{x}, \mathbf{p}), \mathbf{p} \in \mathbf{P}, \quad (8)$$

where MaxN function keeps the top N scores of part detector $\mathbf{p}$ with respect to interaction part $\mathbf{x}$. The per-detector scores are sorted and concatenated as the video representation $\mathbf{f}_v$. Assuming that the traditional max pooling yields a T dimensional feature vector, the Max-N pooling gives us a $T \times N$ feature vector.

To combine the representations from MBHx, MBHy and HOG, we concatenate the $\mathbf{f}_v$ vectors of the three channels as a high dimensional vector. Based on the final video representation, we train a linear SVM classifier. We fix the regularization parameter $C = 10$ for a large dataset, and set $C = 0.01$ for a small dataset to avoid overfitting. We use the LibLinear [5] as our SVM solver.

4. Experiment

4.1. Implementation Details

We evaluate the classification performance of our mid-level interaction part based representation on MPII Cooking [26] and MSR Daily Activity 3D [30] datasets. For our method, during the seeding procedure, we seed 350 interaction parts per-class on MPII Cooking and 100 interaction parts per-class on MSR Daily Activity 3D. During expansion, we expand 10 rounds for MPII Cooking and 5 rounds for MSR Daily Activity 3D. During selection, we fix the per-class part detector number $M = 50$ on MSR Daily Activity 3D. On the large-scale MPII Cooking dataset, 250 part detectors are selected for each action class. We also try different M (per-class part detector number) for parameter evaluation. To form the final video representation, we apply a late fusion approach to concatenate the Fisher vectors of MBHx, MBHy and HOG, and apply Max-N pooling ($N = 2$) to summarize the interaction part scores. These parameters are determined via cross-validation on the training dataset. All the experiments are conducted on a computing server with two Intel Xeon E5450 Quad Core processors (3.00GHz) and 16GB memory.

In the following, we evaluate the classification performance of our mid-level interaction part mining approach on both datasets. We quantitatively compare the algorithmic behavior of our approach under different parameter settings, and compare the performance of our method with the state-of-the-art results.

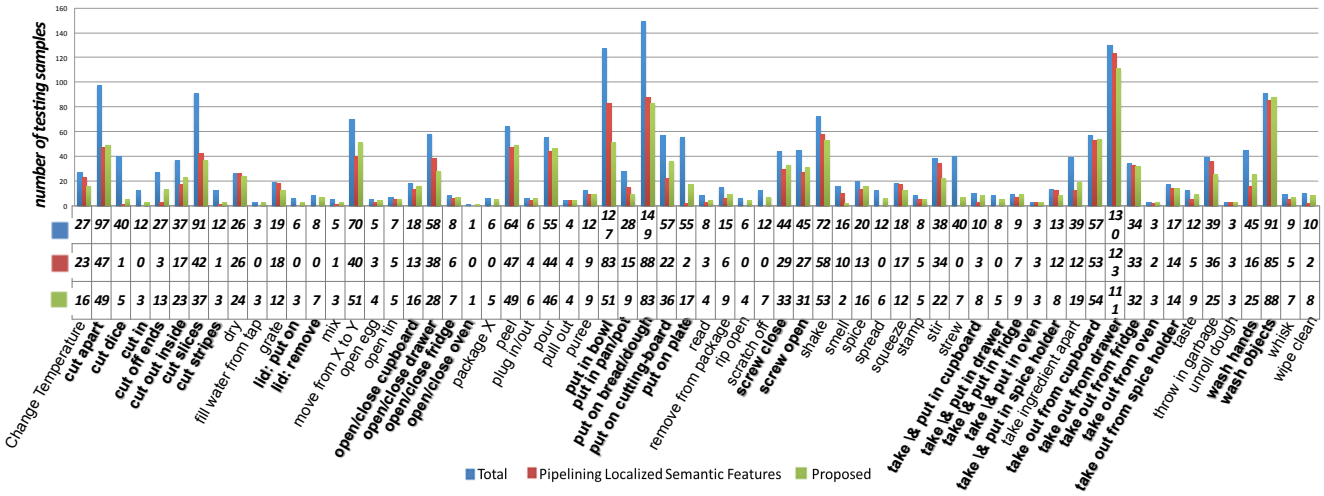


Figure 3: Per-class classification performance comparison on MPII Cooking. We compare our method with the pipelining localized semantic features [38] in terms of classification accuracy, *i.e.*, number of true positive predictions out of total.

Table 1: Classification performance (%) comparison among different methods on MPII Cooking dataset.

Method	Pr	Rc	AP
Pose-based Approach [26]	28.6	28.7	34.6
Holistic Dense Trajectories [29]	49.4	44.8	59.2
Holistic + Pose [26]	50.4	45.1	57.9
Pipelining Localized Semantic Feature [38]	60.1	54.3	70.5
Proposed Method (Max Pooling)	**59.4**	**53.9**	**69.1**
Proposed Method (Max-N pooling)	**62.7**	**55.6**	**72.4**

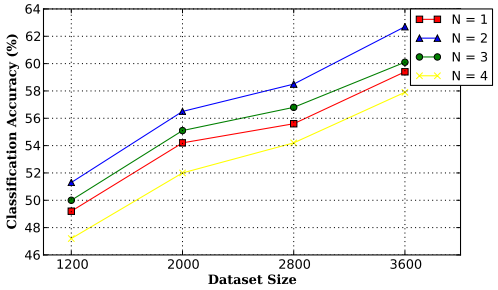


Figure 4: Classification performance under different N for Max-N pooling. The testing is performed by 7 cross-validation rounds on MPII Cooking dataset.

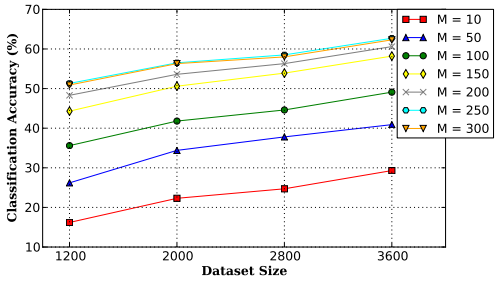


Figure 5: Classification performance by varying the per-class part detector number M. The testing is performed by 7 cross-validation rounds on MPII Cooking dataset.

4.2. MPII Cooking Dataset

MPII Cooking dataset [26] is a very recent fine-grained human cooking action dataset published on CVPR 2012. It is very challenging for fine-grained action recognition due to its large-scale and complexity. A total of 5609 video segments are annotated for 65 action categories such as "open drawer", "cut slices", "cut into dices", "wash hands" or "background" ("background" is dropped in evaluation as indicated in [26]). Following [26], we perform leave-one-subject-out for performance evaluation. 5 out of 12 subjects are used to train the model, and the remaining 7 subjects are tested in 7 cross-validation rounds. We evaluate classification performance in terms of multi-class precision (Pr), recall (Rc) and per-class average precision (AP).

In Table 1, we compare our approach with the state-of-the-art methods on MPII Cooking. First of all, the posed-base approach [26] utilizes human skeleton joints to model human motion, *i.e.*, body model feature and FFT feature. It is observed that the pose-based approach achieves significantly lower performance than the other approaches. This may be because the pose-based approach is based on extremely sparse joint trajectories, which are noisy and coarse. The holistic approach [29] benefits from robust motion features (*i.e.*, MBH, HOG, HOF) around the dense tracks, and achieves much better performance. Marcus *et al.* [26] attempt to combine the low-level features, *i.e.*, pose features and dense motion features, but it only improves the precision from 49.4% to 50.4%. With our mid-level interaction part mining approach, the performance consistently outperforms the low-level feature approaches with more than 10% increase. The pipelining localized semantic feature approach [38] reasonably improves the performance by applying explicit object detection, our approach can achieve

comparable performance with naive max pooling. We can further obtain an increase of more than 2% by applying the novel Max-N pooling. Note that our approach is free from object detection, but achieves better performance.

To demonstrate the algorithmic performance of our approach under different parameter settings, we vary N for Max-N pooling and the per-class part detector number M, which are two important parameters of performance. The results are shown in Figure 4 and Figure 5 respectively. When changing N, the best performance is achieved when $N = 2$, and the naive max pooling strategy ($N = 1$) is not compared to the Max-N pooling ($N = 2$ and $N = 3$). When N continues to increase, there is little improvement. When changing M, the best performance is achieved when $M = 250$. The performance is rather bad when M is very small ($M = 10$ and $M = 50$). This is reasonable because there are far from enough discriminative part detectors being used for a good video representation. Thus the performance is largely improved by increasing M. However, there is little improvement when M reaches the cap ($M = 300$), since it does little help to include the non-discriminative part detectors.

To show the per-class classification performance on MPII Cooking, we compare our method which is free from explicit object detection and the method [38] which requires object detection. The experimental results are shown in Figure 3. The approach of [38] achieves good performance on "put in bowl" and "put on bread/dough" thanks to explicit detection to the bowl and bread/dough, but the object detection accuracy is quite unstable for other objects in their method, *e.g.*, the per-class classification performance is rather bad for some actions such as "lid: remove", "strew", "rip open", "package X", *etc*. Note that both the per-class performance and overall performance of our method are better than [38], which shows the advantage that object detection free method is more stable than the object detection method, not to mention our method saves extensive human labour for object annotation.

To demonstrate the discriminative and representative interaction parts on MPII Cooking, we show some of the top ranked exemplars mined from the action classes of "peel", "put on plate", "wash objects", "wipe clean", "grate" and "stir" in Figure 6. Such exemplars are significant for action representation, *e.g.*, our method discovers the interactions between hands and knives, knives and different vegetables, which are intuitively important for representing the action "peel". For "wipe clean", the interactions between hands and towels, towels and tables are highlighted in the corresponding interaction regions.

4.3. MSR Daily Activity 3D Dataset

MSR Daily Activity 3D dataset [30] is a daily activity dataset captured by a Kinect device, to cover human daily activities in the living room. There are 16 action classes and 10 subjects. Each subject performs an activity in two different poses: "sitting on sofa" and "standing". The total number of the video samples is 320. This dataset is challenging due to the background noise and human-object interactions. We perform cross-subject evaluation to report classification accuracy, *i.e.*, half subjects are used for training and half subjects are tested.

Table 2: Classification performance comparison among different methods on MSR Daily Activity 3D dataset.

Method	Accuracy (%)
Dynamic Temporal Warping [18]	54.0
Moving Pose [37]	73.8
Holistic Dense Trajectories [29]	71.7
Joint Features [30]	68.0
Actionlet Ensemble + Joint Features [30]	85.8
Proposed Method	**83.3**
Proposed Method + Joints Features	**89.3**

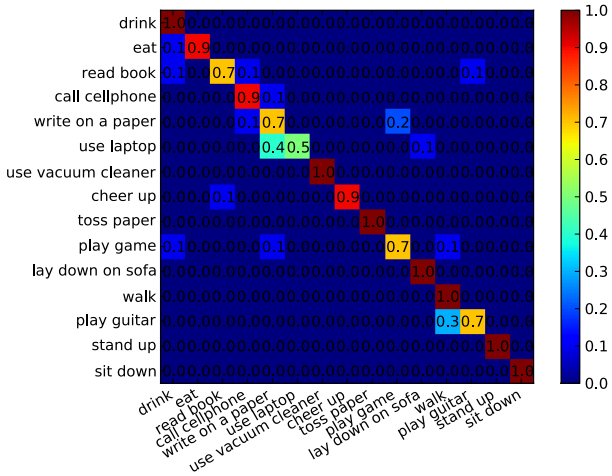
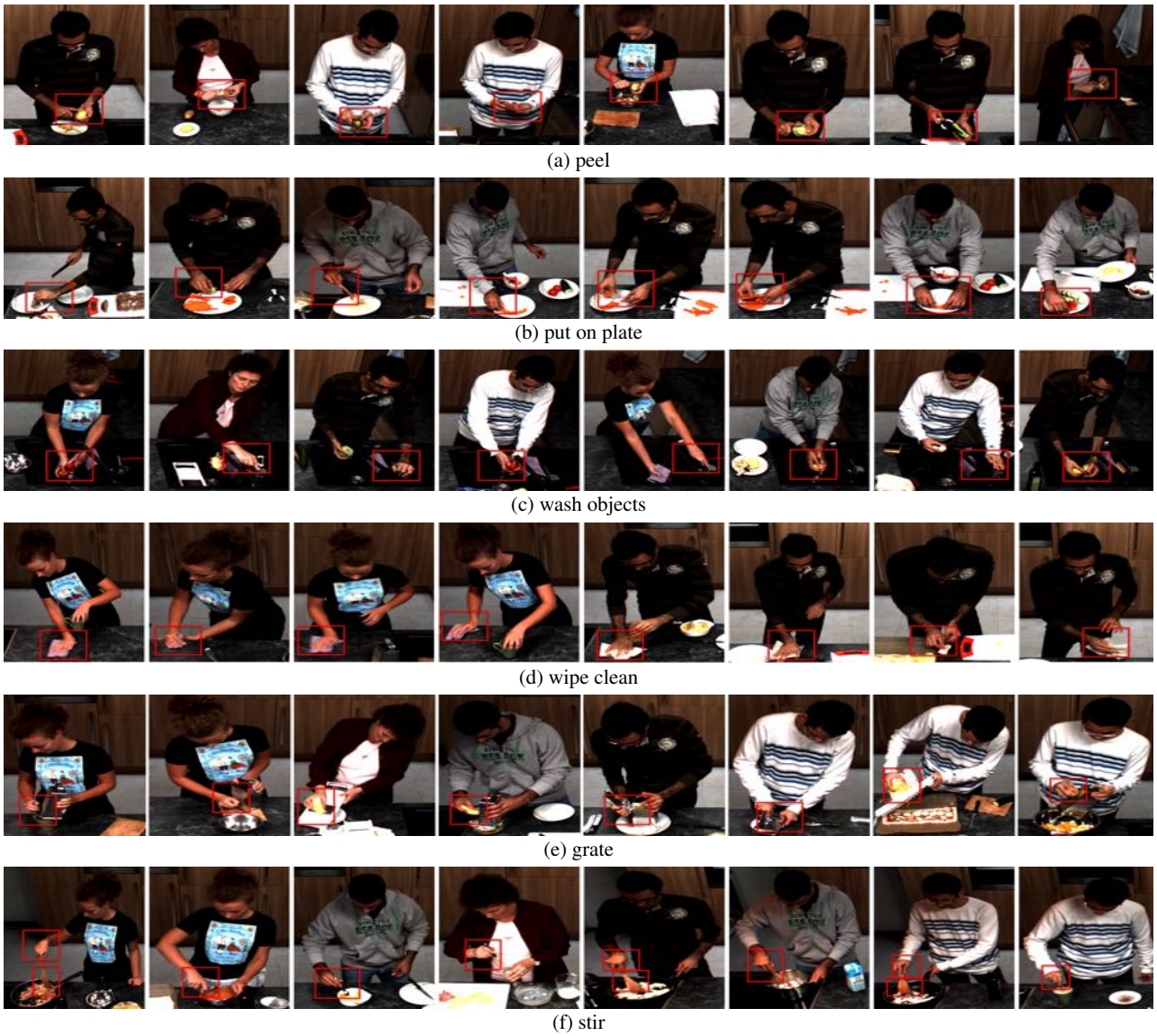


Figure 7: Confusion matrix of using our proposed method on MSR Daily Activity 3D dataset.

We compare our approach with results achieved by the state-of-the-art methods on MSR Daily Activity 3D. The results are shown in Table 2. By only analyzing the 2D video content (*i.e.*, without using 3D skeleton joints and depth map), our approach achieves the classification accuracy of 83.3%, which is much better than the 71.7% by holistic dense trajectory [29] approach. The performance of our approach is improved to 89.3% when we further utilize the skeleton information to localize useful interaction parts and remove background noise. It is still much better than other methods that are based on the skeleton information [18, 37]. Our result is even better than the actionlet ensemble model [30], they obtain 85.8% by using extra depth map, note that we achieve the better performance without using the depth information. Range-sample depth feature [14] achieves 95.6% classification accuracy, however, they use the raw depth information. In contrast, we



(a) peel

(b) put on plate

(c) wash objects

(d) wipe clean

(e) grate

(f) stir

Figure 6: Some exemplars selected from the mined interaction parts. The red boxes represent the interaction regions. For each action class, we apply the most discriminative interaction part detectors to the interaction parts. Then these regions come from the interaction parts which achieve the top detection scores.

only analyze the 2D video content without using the depth map. For fair comparison, we do not include it in Table 2.

Figure 7 shows the confusion matrix of using our approach on MSR Daily Activity 3D. We can observe that the actions containing human-object interactions (*e.g.*, "write on a paper", "use laptop", "play game") are more difficult to be recognized than the other human actions such as "lay down on sofa", "stand up", "sit down". Especially, "use laptop" has the worst recognition accuracy because motions are not salient in such actions and we can only extract relatively sparse interaction parts from the action videos. As long as there exists salient motions, our approach is very effective in recognizing the human-object interactions, even to differentiate some similar actions, *e.g.*, "drink" and "eat".

5. Conclusion

In this paper, we propose an efficient mid-level interaction part mining approach for fine-grained action recognition. In particular, we model human-object interactions without explicit object detection. We validate the proposed approach on two fine-grained action benchmark datasets including MPII Cooking and MSR Daily Activity 3D, and it consistently achieves the state-of-the-art performance on both datasets. Our approach, which is free of explicit object detection, is quite applicable and promising to be applied in fine-grained action recognition applications such as assisted daily living, medical assistance or smart home.

Acknowledgment.
This study is supported by the research grant for the human-centered cyber-physical systems (HCCS) at the Advanced Digital Sciences Center from Singapores Agency for Science, Technology and Research (A*STAR), and supports from ARO grant W911NF-12-1-0057, Faculty Research Awards by NEC Laboratories of America, National Science Foundation of China (NSFC) 61429201.

References

[1] K. Andreev and H. Racke. Balanced graph partitioning. *Theory of Computing Systems*, 39(6):929–939, 2006.

[2] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *CVPR*, 2014.

[3] C. Doersch, A. Gupta, and A. A. Efros. Mid-level visual element discovery as discriminative mode seeking. In *NIPS*, 2013.

[4] V. Escorcia and J. C. Niebles. Spatio-temporal human-object interactions for action recognition in videos. In *ICCV*, 2013.

[5] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *JMLR*, 9:1871–1874, 2008.

[6] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005.

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

[8] A. Jain, A. Gupta, M. Rodriguez, and L. S. Davis. Representing videos using mid-level discriminative patches. In *CVPR*, 2013.

[9] M. Juneja, A. Vedaldi, C. Jawahar, and A. Zisserman. Blocks that shout: Distinctive parts for scene classification. In *CVPR*, 2013.

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

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

[12] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[13] J. Lei, X. Ren, and D. Fox. Fine-grained kitchen activity recognition using rgb-d. In *UbiComp*, 2012.

[14] C. Lu, J. Jia, and C.-K. Tang. Range-sample depth feature for action recognition. In *CVPR*, 2014.

[15] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011.

[16] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009.

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

[18] M. Müller and T. Röder. Motion templates for automatic classification and retrieval of motion capture data. In *SCA*, 2006.

[19] D. Oneata, J. Verbeek, and C. Schmid. Action and event recognition with fisher vectors on a compact feature set. In *ICCV*, 2013.

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

[21] D. J. Patterson, D. Fox, H. Kautz, and M. Philipose. Fine-grained activity recognition by aggregating abstract object usage. In *ISWC*, 2005.

[22] X. Peng, C. Zou, Y. Qiao, and Q. Peng. Action recognition with stacked fisher vectors. In *ECCV*. 2014.

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

[24] A. Prest, V. Ferrari, and C. Schmid. Explicit modeling of human-object interactions in realistic videos. *T-PAMI*, 35(4):835–848, 2013.

[25] M. Raptis, I. Kokkinos, and S. Soatto. Discovering discriminative action parts from mid-level video representations. In *CVPR*, 2012.

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

[27] S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *CVPR*, 2012.

[28] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*. 2012.

[29] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *IJCV*, 103(1):60–79, 2013.

[30] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *CVPR*, 2012.

[31] L. Wang, Y. Qiao, and X. Tang. Mining motion atoms and phrases for complex action recognition. 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] Y. Yang, C. Fermuller, and Y. Aloimonos. Detection of manipulation action consequences (mac). In *CVPR*, 2013.

[35] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei. Human action recognition by learning bases of action attributes and parts. In *ICCV*, 2011.

[36] B. Yao, A. Khosla, and L. Fei-Fei. Classifying actions and measuring action similarity by modeling the mutual context of objects and human poses. *ICML*, 2011.

[37] M. Zanfir, M. Leordeanu, and C. Sminchisescu. The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In *ICCV*, 2013.

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

[39] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *ICPR*, 2004.