Pipelining Localized Semantic Features for Fine-Grained Action Recognition

Yang Zhou[1], Bingbing Ni[2], Shuicheng Yan[3], Pierre Moulin[4], and Qi Tian[1]

[1] University of Texas at San Antonio, USA
[2] Advanced Digital Sciences Center, Singapore
[3] National University of Singapore, Singapore
[4] University of Illinois at Urbana-Champaign, USA
myh511@my.utsa.edu, bingbing.ni@adsc.com.sg, eleyans@nus.edu.sg,
moulin@ifp.uiuc.edu, qi.tian@utsa.edu

Abstract. In fine-grained action (object manipulation) recognition, it is important to encode object semantic (contextual) information, i.e., which object is being manipulated and how it is being operated. However, previous methods for action recognition often represent the semantic information in a global and coarse way and therefore cannot cope with fine-grained actions. In this work, we propose a representation and classification pipeline which seamlessly incorporates localized semantic information into every processing step for fine-grained action recognition. In the feature extraction stage, we explore the geometric information between local motion features and the surrounding objects. In the feature encoding stage, we develop a semantic-grouped locality-constrained linear coding (SG-LLC) method that captures the joint distributions between motion and object-in-use information. Finally, we propose a semantic-aware multiple kernel learning framework (SA-MKL) by utilizing the empirical joint distribution between action and object type for more discriminative action classification. Extensive experiments are performed on the large-scale and difficult fine-grained MPII cooking action dataset. The results show that by effectively accumulating localized semantic information into the action representation and classification pipeline, we significantly improve the fine-grained action classification performance over the existing methods.

1 Introduction

Recently, fine-grained action analysis has raised a lot of research interests due to its potential applications in smart home, medical surveillance, daily living assist and child/elderly care, where action videos are captured indoor with fixed camera. Although background motion (i.e. one of main challenges for general action recognition) is more controlled compared to general action recognition, it is widely acknowledged that fine-grained action recognition (some examples are listed in Figure 8) is very challenging due to large intra-class variability, small inter-class variability, large variety of action categories, complex motions and complicated interactions. Fine-grained actions, especially the manipulation

D. Fleet et al. (Eds.): ECCV 2014, Part IV, LNCS 8692, pp. 481–496, 2014.

sequences involve a large amount of interactions between hands and objects, therefore how to model the interactions between human hands and objects (i.e., context) plays an important role in action representation and recognition. Contextual information has been explored in earlier action recognition works. Feifei et al. [28] modeled objects and human poses jointly by leveraging the mutual context model in human action images. Lan et al. [12,11] introduced the action context descriptor to encode action of individual person and people nearby. Choi et al. [6] proposed to learn crowd action context to recognize collective activities. Marszalek et al. [16] exploited the high correlation between human actions and natural dynamic scenes. Object contextual information has been commonly used for recognizing actions which involves human and object interactions [17,26,24,10]. Feifei et al. [27] jointly modeled the attributes (i.e., actions) and parts (i.e., objects or poselets related to actions) by learning a set of sparse bases that are shown to carry much semantic meaning. However, these methods often represent the human and object contextual information in a global and coarse way, e.g., co-occurrence, which is not sufficient for representing fine-grained actions. This is because in fine-grained actions, local manipulation motion details (e.g., subtle movements of hand in operating an object) are much more important than global co-occurrence information.

The recently proposed local dense motion trajectories [22] has achieved the state-of-the-art performance in action recognition. Local motion trajectory is capable of describing subtle movement, which is suitable for representing fine-grained motion feature. Inspired by this observation, we propose **localized semantic features** (LS) based on local dense motion trajectories. Namely, we extract object occurrence information (i.e., object detection scores) surrounding each local motion trajectory and we augment the semantic features to the motion features. Therefore, we can know which object is being manipulated (object semantic feature) and how it is being manipulated (motion feature) in a localized manner (i.e., per motion trajectory). These complementary information are very important in representing fine-grained actions. Various previous methods have combined semantic features with low-level features for recognition, but they only used global context. For example, Cao et al. [3] only considered grouped feature pooling using global scene type. Chao et al. [5] considered only global label information instead of local semantic.

Further more, we propose a representation and classification pipeline which seamlessly incorporates the localized semantic features into every processing step for fine-grained action recognition. More details are given as follows. In the feature extraction stage, we explore the geometric information between local motion features and the surrounding objects. In the feature encoding stage, we develop a semantic-grouped locality-constrained linear coding (SG-LLC) method that captures the joint distribution between motion and object semantic features. Finally, we propose a semantic-aware multiple kernel learning (SA-MKL) framework by utilizing the empirical joint distributions between action and object type for more discriminative action classification. The proposed pipeline is experimented thoroughly on the fine-grained MPII cooking action dataset [20], which is the

Name	Color	Name	Color
1.background	⬜	17.oven	🟥
2.bottle	🟥	18.food wrapper	🟥
3.bowel	🟦	19.pan	🟥
4.bread	🟦	20.slicer(for grate)	🟪
5.plug-out charger	🟩	21.plate	🟧
6.electric range	🟩	22.pot	🟧
7.cup(transparent)	🟩	23.electric blenders	🟥
8.cupboard	🟦	24.small size food	🟪
9.cuttingboard	🟩	25.seasoning bottle	⬛
10.dough	🟩	26.bottle rack	⬛
11.drawer	🟩	27.hands-on juicers	🟦
12.eggs	🟦	28.tin	🟦
13.fridge	🟥	29.tin opener	🟨
14.hands	🟥	30.towel	🟨
15.lid	🟥	31.water sink	🟨
16.small objects	🟪	32.human body	🟩

Fig. 1. Color code for 32 types object-of-interest

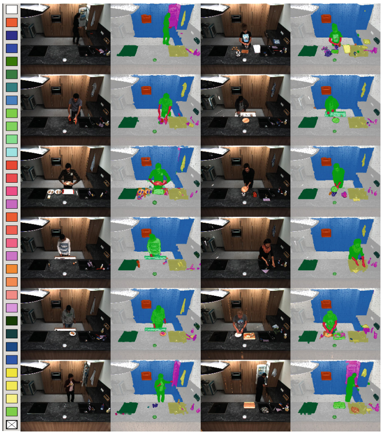


Fig. 2. Sample object detection maps

large-scale and very challenging dataset for fine-grained action recognition. The results show that the localized semantic action representation and classification pipeline can step-by-step improve the action classification performance, which significantly outperforms the existing methods on the MPII cooking dataset in terms of multi-class precision, recall and per-class average precision.

To summarize, our contributions are three-fold: 1) we propose an end-to-end solution on utilizing **localized semantic** features (i.e., object contextual information of **local** dense trajectories in the spatial-temporal volume) in fine-grained action analysis, which includes novel **localized semantic** feature encoding, pooling and classification; 2) we propose a novel MKL modeling and optimization framework for **semantic-aware** classifier learning, which utilizes the prior knowledge of kernel weights; 3) the proposed fine-grained action recognition pipeline achieves about 10% improvement over the existing methods on the challenging fine-grained action dataset.

2 Methodology

2.1 Localized Semantic Feature Extraction

As introduced in the previous section, our basic idea is to augment local motion features with *localized semantic features* (LS), to enrich the descriptions for representing manipulation movement that involves subtle human and object interactions. To this end, we first extract local dense motion trajectories [22] from input videos. To describe motion, different types of motion feature descriptors are computed in a spatial-temporal volume (i.e., spatial size of 2 × 2 with temporal length of 15) around the 3D neighborhood of the tracked points along the trajectory. Following [22], we use four types of motion feature descriptors in-

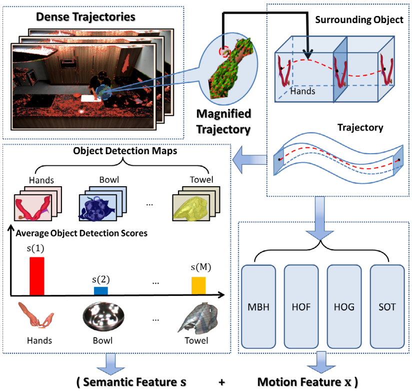


Fig. 3. Localized semantic feature extraction

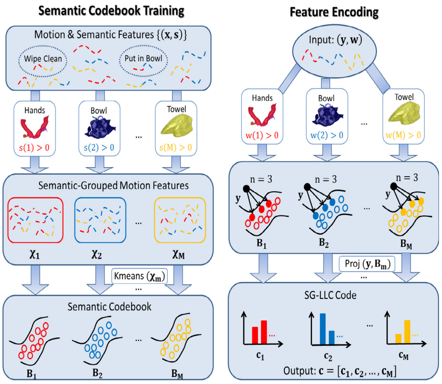


Fig. 4. Semantic-grouped (best viewed with color) feature encoding

cluding histogram of oriented gradients (HOG), histogram of optic flows (HOF), motion boundary histogram (MBH) and shape of trajectory (SOT).

During human-object interaction, local motion features describe *how a certain object is being manipulated*. For fine-grained actions, different action types share similar motion patterns, for example, local motions are almost the same among actions "put on plate", "put on pot", "put on dough". Therefore, we should augment each local motion feature with *localized semantic feature* to encode *which object is being manipulated*, i.e., whether the action is related to "plate", "pot" or "dough"? In other words, the localized semantic feature descriptor encodes the local object-in-use contextual information for each local motion trajectory. To compute localized semantic features, for each input video frame, we first build object detection maps for various types of objects. Assume we have M objects of interest, then each position on the detection map is represented by a M-dimensional object detection score vector. For a trajectory, we average the object detection score vectors in the spatial-temporal volume along its tracked points and form a M-dimensional localized semantic feature vector.

Object detection maps are computed as follows. We first apply superpixel segmentation using SLIC [1] on each input frame. The 1624×1224 pixels video frame is over-segmented into around 2000 superpixels, with the regularization parameter being set as 10. We then represent each superpixel with a concatenated feature vector consisting of histogram of oriented gradient (HOG) [7] and HSV color histogram. Multiple linear support vector machine classifiers are applied to calculate the object detection scores. We build our training object patch (superpixel) dataset by randomly sampling 12000 video frames from the training videos. In average, we have annotated around 2000 training patches for each object type. In addition, we use the conditional random field model to spatially regularize the object detection map for better detection accuracy. For the MPII cooking dataset, 32 types of object-of-interest are defined in data-driven approach, which are summarized in Figure 1, some object detection maps are shown in Figure 2. The feature extraction process is illustrated in Figure 3.

Discussion: One might argue that pre-detection of objects gives *unfair* advantages to our method over conventional holistic action recognition method [22]. We clarify that: 1) we do not target general action recognition problem on action datasets such as YouTube [15], Hollywood2 [16], UCF sport [25], etc., where object detection is infeasible. Instead, fine-grained action recognition is more suitable for applications such as indoor assisted living, occupational therapy (with fixed camera), where object detection is quite feasible. Indeed, to detect object-of-interest is compulsory task in these applications; 2) for fine-grained actions with frequent and delicate hand-object interactions, to detect object and associate it locally with motion features is a natural, reasonable and promising approach. Holistic approaches such as bag of dense trajectories [22] or STIPs [13] cannot well deal with fine-grained action recognition, even though their implementations are easier without the need for object detection.

2.2 Semantic-Grouped Feature Encoding

The next important building block for image and video classification is local feature encoding. State-of-the-art local feature encoding schemes include vector quantization (or bag-of-words) [8], locality-constrained linear coding (LLC) [23], fisher kernel [18], etc. Usually, a codebook is trained using the training features, any input feature vector can be encoded using the codebook either by searching its nearest codebook item (visual word) or by computing the linear combination of codebook items that well approximates it (i.e., LLC).

In this work, each local motion feature is augmented with a localized semantic feature vector, which indicates which object(s) the motion feature is associated with. This contextual information motivates us to propose an enhanced feature encoding scheme. The basic idea is as follows. As the localized semantic features tell us to which object(s)-of-interest the motion feature is related, when we encode a local motion feature descriptor, we should encourage that the codebook items it selects are also related to the same object(s)-of-interest. We believe that the advantages of this localized semantic feature grouped feature encoding are two-fold: firstly, it implicitly embeds the information of *which object(s) is being manipulated* into the encoded feature representation; secondly, because the codebook motion features that are related to the same object(s)-of-interest are considered for approximating the input motion feature, the similarity between the input motion feature and the selected codebook items is higher, thus more accurate encoding (i.e., lower reconstruction error) can be achieved. The proposed semantic-grouped feature encoding is illustrated in Figure 4 and more details are introduced as follows.

We denote by $(\mathbf{x}, \mathbf{s})$ the pair of motion descriptor $\mathbf{x}$ and the corresponding localized semantic feature vector $\mathbf{s}$. Namely, $\mathbf{x}$ represents the concatenation of HOG, HOF, MBH and SOT feature descriptors and $\mathbf{s}$ is a M-dimensional object detection score vector. Let $\mathcal{X}$ be a set of features extracted from the training video clips, i.e., $\mathcal{X} = \{(\mathbf{x}^1, \mathbf{s}^1), \cdots, (\mathbf{x}^N, \mathbf{s}^N)\}$. The total number of training features is N. According to the localized semantic features $\{\mathbf{s}^i\}$, we further group the whole training feature set $\mathcal{X}$ into M subsets $\mathcal{X} = \bigcup \mathcal{X}_m, m = 1, \cdots, M$. Each $\mathcal{X}_m$ only contains the set of features that are related to object-of-interest type m, i.e., $\mathbf{s}(m) > 0$, we denote by $\mathbf{s}(m)$ the m-th element of vector $\mathbf{s}$. Note that one feature can be related to multiple objects-of-interest (i.e., the trajectory is surrounded by multiple objects), therefore different $\mathcal{X}_m$ may be overlapped. For each $\mathcal{X}_m$, we use the K-means clustering algorithm to train a codebook of motion features B_m. Note that each B_m is a $D \times N_m$ matrix, i.e., D is the motion feature dimension and N_m is the number of basis for codebook B_m. We denote by B_0 the codebook trained using the whole training set $\mathcal{X}$. Therefore our codebook can be denoted as $B = [B_0, B_1, \cdots, B_M]$. Each sub-codebook $B_m, m = 1, \cdots, M$ is related to m-th type of object-of-interest.

Given an input feature descriptor $(\mathbf{y}, \mathbf{w})$, i.e., local motion feature $\mathbf{y}$ and localized semantic feature $\mathbf{w}$ vector pair, the encoding objective is to minimize the following cost function with respect to encoding coefficients $\mathbf{c} = [\mathbf{c}_0; \mathbf{c}_1; \cdots; \mathbf{c}_M]$:

$$\min_{\mathbf{c}} \|\mathbf{y} - [B_0, B_1, \cdots, B_M][\mathbf{c}_0; \mathbf{c}_1; \cdots; \mathbf{c}_M]\|_2^2 \tag{1}$$

$$s.t. \quad \sum_{m=0}^{M} |\mathbf{c}_m| \leq \varepsilon, \quad \varepsilon > 0, \tag{2}$$

$$\sum_{m=1}^{M} (1 - \mathrm{w}_m) |\mathbf{c}_m| \leq \tau, \quad \tau > 0, \tag{3}$$

here $\mathbf{c}_m$ is the encoding coefficient on sub-codebook B_m. The first constraint Eqn. (2) encourages that: 1) only a few sub-codebooks are selected for reconstructing the input local motion feature vector $\mathbf{y}$ and 2) the codebook items are sparsely selected. The second constraint Eqn. (3) encourages that the sub-codebooks which are not related to the motion feature $\mathbf{y}$ (i.e., the sub-codebook

Algorithm 1. Semantic-grouped locality-constrained linear coding

```
input: feature descriptor pair $(\mathbf{y}, \mathbf{w})$, number of nearest neighbors $n$,
regularization term $\beta$ of sparse coding solver, sub-codebooks $B_1, \cdots, B_M$.
Initialize $\tilde{B} = [\ ]$, $\beta = 500$, $n = 5$, compute $\mathbf{c}_0$ with LLC encoding on $B_0$.
for $m = 1, \cdots, M$ do
    if $\mathbf{w}(m) > 0$
        Choose n nearest neighbors of $\mathbf{y}$ from $B_m$ as $\tilde{B}_m$.
        Push $\tilde{B}_m$ into $\tilde{B}$, i.e., $\tilde{B} = [\tilde{B}, \tilde{B}_m]$.
    else
        $\mathbf{c}_m = 0$.
end
Solve $\mathbf{c}$ following sparse coding solver in [23]:
    (1) $\tilde{\mathbf{c}} = \mathbf{C} + \beta \mathrm{diag}(\mathbf{C}) \backslash \mathbf{1}$, where $\mathbf{C} = (\tilde{B} - \mathbf{1}\mathbf{y}^{\mathrm{T}})(\tilde{B} - \mathbf{1}\mathbf{y}^{\mathrm{T}})^{\mathrm{T}}$.
    (2) $\mathbf{c} = \tilde{\mathbf{c}} / \mathbf{1}^{\mathrm{T}} \tilde{\mathbf{c}}$,
Assign $\mathbf{c}$ to the corresponding positions of $\mathbf{c}_m$, i.e., $\mathbf{w}(m) > 0$.
output: SG-LLC code $\mathbf{c} = [\mathbf{c}_0; \mathbf{c}_1; \cdots; \mathbf{c}_M]$.
```

m that $\mathbf{w}(m)$ is near zero) are not selected for reconstructing $\mathbf{y}$. Applying these two constraints ensures that only a few codebook items which are related to the object-in-use of the motion feature $\mathbf{y}$ are selected for approximating the input motion feature vector $\mathbf{y}$.

Finding the exact solution to Eqn. (1) is possible by various sparse coding solvers including generic QP solvers (e.g., CVX), ℓ^1-regularized Least Squares solver [14], etc. However, for a large-scale dataset as MPII cooking, exact solution for encoding millions of local motion features is extremely expensive. In practice, we develop an approximate optimization algorithm which is shown in Algorithm 1. The basic idea is to first choose n nearest codebook items from the sub-codebooks which are selected by non-zero semantic scores of the input motion feature, and then perform reconstruction using the codebook items. We note that this approximated optimization for feature encoding is similar to the locality-constrained linear coding algorithm (LLC). Therefore, our feature encoding algorithm is named as *Semantic-Grouped Locality-constrained Linear Coding (SG-LLC).*

2.3 Semantic-Aware Motion Feature Pooling and Classification

After local motion feature encoding, the next building blocks for action classification are to perform local motion feature pooling and classifier learning. Feature pooling is to aggregate local features to form a video level representation (i.e., to form a representation vector). In this subsection, we show how localized semantic features can help enhance the pooling and classifier training stage, i.e., to achieve more discriminative video level representation and classification. The procedure is shown in Figure 5.

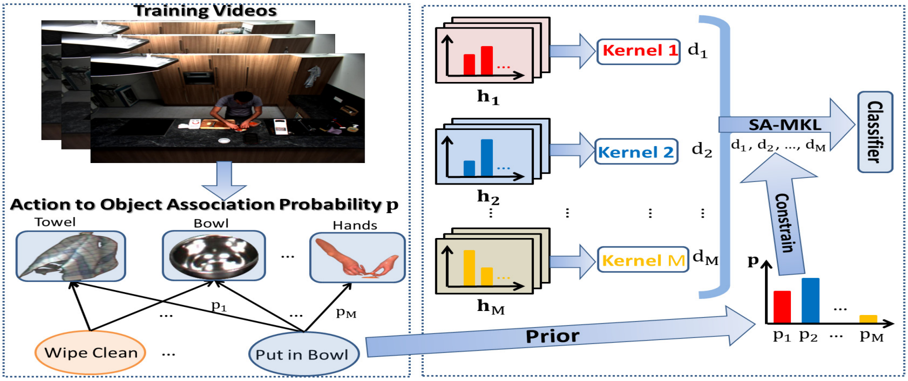


Fig. 5. Semantic-Aware multiple kernel learning (SA-MKL) utilizing action to object association probability (Eqn. (4)) as prior information

Semantic-Partitioned Feature Pooling: A traditional motion feature pooling scheme is global pooling, i.e., a frequency vector of all encoded local motion features within the video volume is calculated, named as the bag-of-words frequency vector representation $\mathbf{h}$. With localized semantic feature vector for each local motion feature, we can perform *finer* pooling. More specific, global pooling ignores the object-in-use contextual information and the local motion features associated with different irrelevant objects are pooled together, therefore, the resulting histogram representation is noisy (i.e., confused by motion features occurring on different objects) and less discriminative. On the contrary, if we utilize the localized semantic information, we can pool the local motion features *object-wise.* Namely, local motion features that are associated with the same object are pooled together and we can have multiple pooled histogram representations where each corresponds to the distribution of motions related to one type of object. It is obvious this new histogram representation possesses richer and finer descriptive information than globally pooled histogram.

The proposed pooling process is as follows. Suppose for video volume V, we have a set of extracted local motion features $\mathcal{X} = \{(\mathbf{x}^1, \mathbf{s}^1), \cdots, (\mathbf{x}^N, \mathbf{s}^N)\}$. Each local feature $\mathbf{x}$ is encoded as $\mathbf{c}$. According to localized semantic features we can group (partition) the local motion feature set $\mathcal{X}$ into M subsets $\mathcal{X} = \bigcup \mathcal{X}_m, m = 1, \cdots, M$, where each $\mathcal{X}_m$ only contains motion features which are associated with m-th object-of-interest, i.e., $\mathbf{s}(m) > 0$. We then calculate the pooled vector (histogram) within each $\mathcal{X}_m$ and result in M histogram vectors as $\{\mathbf{h}_1, \cdots, \mathbf{h}_M\}$. We also denote $\mathbf{h}_0$ as the pooled histogram vector using all local motion features in the video volume, i.e., $\mathcal{X}$.

Semantic-Aware Multiple Kernel Learning: Now we have $M+1$ feature channels for each video clip (i.e., each feature channel corresponds to one object-associated histogram $\mathbf{h}_m$, $m = 0, 1, \cdots, M$), a straightforward feature fusion and classification scheme is to calculate $M+1$ kernel matrices $\{\mathbf{K}_0, \mathbf{K}_1, \cdots, \mathbf{K}_M\}$ and combine them for classifier training. There are two major kernel combination ways include: 1) average kernel combination [21]; and 2) kernel weights learning, i.e., multiple kernel learning [2]. Traditional multiple kernel learning methods do not rely on any prior knowledge about the kernel weights $\{d_1, \cdots, d_M\}$, i.e., the value of d_m means how important kernel $\mathbf{K}_m$ is. However, for our problem, as each type of action is strongly correlated with certain types of objects, prior knowledge on the kernel weights are available. For example, to recognize the action "put in bowl", the kernels related to object "hands" and "bowl" are important. To take advantage of the prior knowledge brought by the localized semantic feature, we therefore propose a novel multiple kernel learning method which can leverage the empirical joint distributions between action and object type. Namely, the empirical action-object association probability estimated from the training data guides the learning of kernel weights $\{d_1, \cdots, d_M\}$.

To begin with, we define the empirical action-object association probability for action a (we ignore the superscript for action a in the rest of paper) and object m as:

$$p_m^a = \frac{\sum_{i=1}^{N_{tr}} \left(y_i = 1 \wedge \mathbf{h}_m^i > \mathbf{0}\right)}{\sum_{i=1}^{N_{tr}} (y_i = 1)}, \; m = 1, \cdots, M, \tag{4}$$

here N_{tr} denotes the number of training video clips, and $y_i \in \{+1, -1\}$ denotes binary classification label for video level representation $\mathbf{h}^i$. The numerator represents the number of training video clips which have action label a and there is object-use on the m-th object. The denominator denotes the number of positive training video clips for action a.

We consider one-versus-all classification in this work. 64 action types are defined in dataset, and a total of 64 binary classifiers $f(\mathbf{h})$ are learned. For each binary classifier (i.e., to classify action a), we define the following decision function:

$$f^a(\mathbf{h}) = f_0^a(\mathbf{h}) + \Delta f^a(\mathbf{h}), \tag{5}$$

here $f_0^a(\mathbf{h}) = \mathbf{w}_0^{\mathrm{T}} \phi_0(\mathbf{h}_0) + b$ is the base classifier trained from the globally pooled histogram vector $\mathbf{h}_0$. $\Delta f^a(\mathbf{h})$ is a linear combination of object-specific classifiers learned from their corresponding object-specific histogram vector $\mathbf{h}_m, m = 1, \cdots, M$, which is defined as in Eqn. (6):

$$\begin{aligned} \Delta f^a(\mathbf{h}) &= \sum_{m=1}^{M} d_m \mathbf{w}_m^{\mathrm{T}} \phi_m(\mathbf{h}_m) + b \\ s.t. \quad & \mathbf{d} \geq \mathbf{0}, \quad ||\mathbf{d}||_\infty \leq 1, \end{aligned} \tag{6}$$

where $\mathbf{d} = [d_1, \cdots, d_M]^{\mathrm{T}}$ are the weights for combining different classifiers. The combined classifier can be learned by optimizing the following objective function:

$$\begin{aligned} \min_{d_m} \min_{\mathbf{v}_{m,b,\xi_i}} \; & \frac{1}{2} \sum_{m=1}^{M} \frac{\|\mathbf{v}_m\|}{d_m} + \frac{\lambda}{2} \sum_{m=1}^{M} |d_m - p_m| + C \sum_{i=1}^{N_{tr}} \xi_i \\ s.t. \; y_i & \left(\mathbf{w}_0^{\mathrm{T}} \phi_0(\mathbf{h}_0^i) + \sum_{m=1}^{M} \mathbf{v}_m^{\mathrm{T}} \phi_m(\mathbf{h}_m^i) + b \right) \geq 1 - \xi_i, \\ & \xi_i \geq 0, i = 1, \cdots, N_{tr}, \quad \mathbf{d} \geq \mathbf{0}, \quad ||\mathbf{d}||_\infty \leq 1, \end{aligned} \tag{7}$$

where we set $C = 100$ as the multiple kernel learning regularization parameter. $\mathbf{K}_m(\mathbf{h}_m^i, \mathbf{h}_m^j) = \phi_m(\mathbf{h}_m^i)^{\mathrm{T}} \phi_m(\mathbf{h}_m^j)$. p_m is the action-object association probability for object m, which is defined in Eqn. (4). $\lambda, d_1, \cdots, d_M$ are the parameters we need to learn. Note that the second objective, i.e., $|d_m - p_m|$ enforces that the kernel weights to approximate the values of action-object association probability p_m. λ adjusts the weight between kernel $\mathbf{K}_0$ and semantic kernels $\mathbf{K}_1, \cdots, \mathbf{K}_M$. Large λ will encourage that the learned object-specific kernel weight follows the empirical action-object association probability.

To solve the objective Eqn. (7), we alternatively optimize w.r.t. the variables $d_m, \mathbf{v}_m, b, \xi_i$ using the following two steps.

Firstly, we optimize $\mathbf{v}_m$, b, ξ_i with fixed d_m. By introducing the non-negative Lagrangian multipliers $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_{N_{tr}}]^{\mathrm{T}}$, the dual can be derived as follows:

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} \boldsymbol{\alpha}^{\mathrm{T}}\mathbf{1} - \frac{1}{2}(\boldsymbol{\alpha} \odot \mathbf{y})^{\mathrm{T}}(\sum_{m=1}^{M} d_m \mathbf{K}_m + \mathbf{K}_0)(\boldsymbol{\alpha} \odot \mathbf{y}) \\ &s.t. \quad \boldsymbol{\alpha}^{\mathrm{T}}\mathbf{y} = 0, \quad 0 \leq \boldsymbol{\alpha} \leq C, \end{aligned} \tag{8}$$

where $\boldsymbol{\alpha} \odot \mathbf{y}$ denotes the element-wise product between two vectors $\boldsymbol{\alpha}$ and $\mathbf{y}$. Because Eqn. (8) is a standard dual problem, we can solve it with the SVM solvers such as libsvm [4]. With the dual primal coefficients $\boldsymbol{\alpha}$ derived from the SVM solvers, we compute the primal variables $\mathbf{v}_m$ as:

$$\mathbf{v}_m = d_m \sum_{i=1}^{N_{tr}} \alpha_i y_i \phi_m(\mathbf{h}_m^i), m = 1, \cdots, M. \tag{9}$$

Secondly, we optimize d_m with fixed $\mathbf{v}_m, b, \xi_i$, the problem in Eqn. (7) reduces to:

$$\begin{aligned} &\min_{d_m} \frac{1}{2} \sum_{m=1}^{M} \frac{\|\mathbf{v}_m\|}{d_m} + \frac{\lambda}{2} \sum_{m=1}^{M} |d_m - p_m| \\ &s.t. \quad \mathbf{d} \geq \mathbf{0}, \quad ||\mathbf{d}||_\infty \leq 1. \end{aligned} \tag{10}$$

By taking the derivative over d_m, the closed-form solution is given in Eqn. (11):

$$d_m = \max\{\sqrt{\frac{\|\mathbf{v}_m\|}{\lambda}}, p_m\}. \tag{11}$$

The optimization procedure is given in Algorithm 2. Finally, the classifier for a novel input $\mathbf{h} = \{\mathbf{h}_0, \mathbf{h}_1, \cdots, \mathbf{h}_M\}$ is expressed as Eqn. (12):

$$f^a(\mathbf{h}) = \sum_{i=1}^{N_{tr}} \alpha_i y_i \left[\sum_{m=1}^{M} d_m \mathbf{K}_m(\mathbf{h}, \mathbf{h}^i) + \mathbf{K}_0(\mathbf{h}, \mathbf{h}^i) \right] + b. \tag{12}$$

Algorithm 2. Optimization for Semantic-Aware Multiple Kernel Learning

```
input: $\mathbf{d}^0$, $\lambda$, $\epsilon$, $\{\mathbf{K}_0, \mathbf{K}_1, \cdots, \mathbf{K}_M\}$, $\{p_1, \cdots, p_M\}$
Initialize $d_m^0 = 1/M$ $(m = 1, \cdots, M)$, $\lambda = 0.2$, $\epsilon = 10^{-4}$.
repeat
    Compute $\boldsymbol{\alpha}^t$ by solving Eqn. (8) using SVM solver with $\mathbf{d}^t$.
    Compute $\mathbf{v}_m$ by Eqn. (9) and solve $\mathbf{d}^{t+1}$ by Eqn. (11).
    t=t+1.
until $\|\mathbf{d}^{t+1} - \mathbf{d}^t\| < \epsilon$
output: $\boldsymbol{\alpha}$, $\mathbf{d}$, $\lambda$
```

3 Experiment

3.1 Dataset and Configurations

We perform extensive experiments on the MPII cooking [20] dataset, which is a recent fine-grained cooking action dataset published on CVPR 2012. Considering the scale and complexity, it is very challenging for fine-grained action recognition.

Totally, 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 [20]). Following the same experimental setting as in [20], 5 out of 12 subject are used to train the model, the remaining 7 subjects are used to perform leave-one-person-out cross-validation. We evaluate classification performance in terms of multi-class precision (Pr), recall (Rc) and per-class average precision (AP).

For codebook training, the base codebook $\mathbf{B}_0$ is clustered into 4000 centers, all the other object-specific codebooks have 500 cluster centers. For the original holistic bag-of-words on dense motion trajectories method [22], the size of codebook is also set as 4000 for all types of descriptors for fair comparison. All the experiments are conducted on a powerful 16-core computing server. Each step is paralleled if applicable, and our pipeline (with object detection) involves 9 hours of running time in total.

In the following, we first evaluate the effectiveness of every component of our proposed localized semantic feature based fine-grained action recognition pipeline, which includes both semantic-grouped feature encoding and semantic-aware multiple kernel learning. Then we quantitatively compare the classification performance of our method with state-of-the-art results on the MPII cooking dataset with in-depth discussions on the algorithmic behavior of our approach.

3.2 Results and Discussions

Effectiveness of Semantic-Grouped Feature Encoding: We show the effectiveness of proposed SG-LLC in Table 1. We compare various state-of-the-art encoding methods including: vector quantization encoding (VQ), locality-constrained linear coding (LLC) and our proposed semantic-grouped locality-

Table 1. Comparison among different encoding methods in terms of multi-class precision (%)

	VQ	LLC	SG-LLC
HOG	39.6	42.2	46.2
HOF	41.3	42.8	45.7
MBHx	42.4	44.9	49.3
MBHy	45.6	47.1	51.8
SOT	39.2	42.3	47.6
Combined	49.4	52.5	57.3

Table 2. Comparison among different multiple kernel learning methods in terms of multi-class precision (%)

	AK-SVM	MKL	SA-MKL
HOG+SG-LLC	46.2	47.1	48.7
HOF+SG-LLC	45.7	46.9	48.3
MBHx+SG-LLC	49.3	50.5	52.4
MBHy+SG-LLC	51.8	53.1	54.7
SOT+SG-LLC	47.6	47.9	49.3
Combined+SG-LLC	57.3	58.2	60.1

constrained encoding (SG-LLC). To be comprehensive, these encoding techniques are tested on individual feature descriptor and their combination (Combined). From the comparison results shown in Table 1, we observe that the SG-LLC coding consistently and significantly enhances the discriminative power for all types of motion feature descriptors as well as their combination. More specific, SG-LLC is much more discriminative than LLC (i.e., which does not consider localized semantic information), and the improvement from LLC to SG-LLC is over 5% for most feature descriptors. This demonstrates that the encoding method to embed localized semantic information into motion feature encoding is beneficial. We also study our algorithmic performance by varying the number of nearest neighbors parameter n for our algorithm SG-LLC, i.e., $n = 2, 5, 20, 40$. As illustrated in Figure 6, $n = 5$ gives the best performance, and larger n will induce more noise and decrease classification performance.

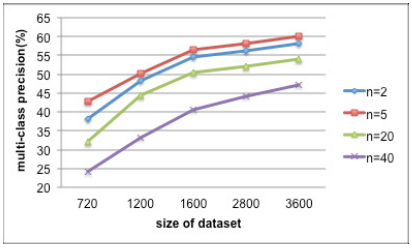


Fig. 6. Classification test on 7 cross-validation rounds under different n

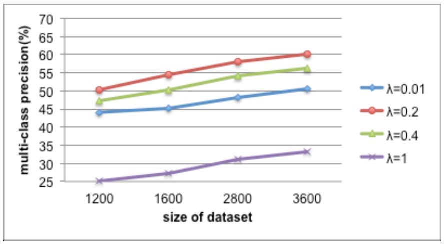


Fig. 7. Classification test on 7 cross-validation rounds under different $\boldsymbol{\lambda}$

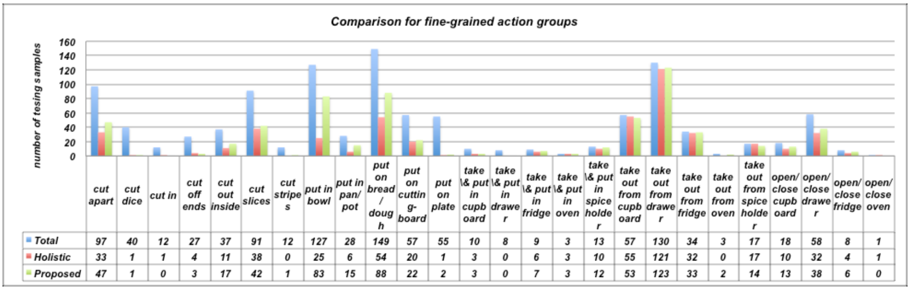
Effectiveness of Semantic-Aware Motion Feature Pooling and Classification: We show in Table 2 that after semantic-grouped feature encoding (SG-LLC), our proposed semantic-aware motion feature pooling and multiple kernel learning (SA-MKL) can further boost the classification performance. To this end, Table 2 compares our proposed SA-MKL with conventionally used average kernel (AK-SVM) in action recognition [22] as well as conventional multiple kernel learning method (MKL). For MKL, we use the state-of-the-art implementation of SimpleMKL [19]. We test on different types of motion features (which are encoded by the proposed SG-LLC method) and the results show that 1) MKL outperforms average kernel due to its kernel selection capability and 2) our proposed SA-MKL further outperforms traditional MKL as our SA-MKL method utilizes prior information for the kernel weights through semantic information extraction for kernel learning, i.e., action class-object type contextual information. We also study the effect of the parameter λ used for adjusting the weight between kernel $\mathbf{K}_0$ and kernels $\mathbf{K}_1, \cdots, \mathbf{K}_M$. Figure 7 illustrates the classification performance by varying λ. As can be seen from Figure 7, small λ improves classification performance, which is benefited from prior semantic information, and $\lambda = 0.2$ achieves the best result. Performance starts to drop from $\lambda = 0.4$ because of the magnified semantic noise.

Table 3. Stage-by-stage classification performance(%) of our proposed pipeline

	Pr	Rc	AP
Holistic Dense Trajectories [22]	49.4	44.8	59.2
Holistic + Pose [20]	50.4	45.1	57.9
Dense Trajectory + LS	53.9	48.9	64.4
SG-LLC + AK-SVM	57.3	52.4	68.4
SG-LLC + SA-MKL	**60.1**	**54.3**	**70.5**

Comparison with the State-of-the-Art: We compare our approach with the state-of-the-art performance achieved by the holistic dense motion trajectory approach [22] (naive combination of motion features with pose features is used in [20], which achieves minor improvement). To study the algorithmic behavior of our pipeline (i.e., to show the stage-by-stage improvement of the pipeline), we also compare our method with: 1) naive combination of the dense motion trajectory bag-of-words features and the localized semantic bag-of-words features (Dense trajectory + LS, Average Kernel is used) and 2) our proposed SG-LLC encodings but without semantic-aware pooling and multiple kernel learning (SG-LLC + Average Kernel). Comparison results are shown in Table 3. In our experiment, we set λ and n to be 0.2 and 5 empirically. The results show that naive combination of local motion features and localized semantic features improves the holistic dense trajectory method. However, by exploring novel ways to embed the localized semantic features into feature coding, pooling and classification steps, we can obtain a total of more than 10% performance increase accumulated by every stage of our proposed pipeline, which is much better than merely using the semantic feature and combining it naively with the original motion feature descriptors (about 6% more increase than naive combination of dense trajectory and LS). Also each proposed step (i.e., SG-LLC and SA-MKL) consistently benefits the final fine-grained action classification performance.

To prove the effectiveness of our approach on fine-grained actions, we specifically pick up classification results of five fine-grained action groups (i.e., "cut", "put in/on", "take & put in", "take out", "open/close") and compare our approach with holistic dense trajectories in Figure 8, we observe that our method significantly outperforms the holistic approach on the fine-grained action recognition. We find that recognition on actions of "put in/on" have been significantly improved, which are benefited from excellent object detection performance on objects such as bowel, bread/dough or cutting-board (i.e., manipulated objects in the "put in/on" video clips). However, actions of "put on plate" are not improved as expected because the plate is always occluded and difficult for detection. We also observe that "cut" actions are not improved significantly compared to "put in/on", the reasons can be two-fold: 1) the intra-class variability is especially large and 2) the object detection is extremely difficult for the manipulated objects (e.g., knife, fruits, vegetables) because they are in very small size. Nevertheless, "cut" actions are still improved by incorporating the localized semantic information into further steps of our pipeline.



	cut apart	cut dice	cut in	cut off ends	cut out inside	cut slices	cut stripes	put in bowl	put in pan/ pot	put on bread / dough	put on cutting-board	put on plate	take \& put in cupboard	take \& put in drawer	take \& put in fridge	take \& put in oven	take \& put in spice holder	take out from cupboard	take out from drawer	take out from fridge	take out from oven	take out from spice holder	open/ close cupboard	open/ close drawer	open/ close fridge	open/ close oven
🟦 Total	97	40	12	27	37	91	12	127	28	149	57	55	10	8	9	3	13	57	130	34	3	17	18	58	8	1
🟥 Holistic	33	1	1	4	11	38	0	25	6	54	20	1	3	0	6	3	10	55	121	32	0	17	10	32	4	1
🟩 Proposed	47	1	0	3	17	42	1	83	15	88	22	2	3	0	7	3	12	53	123	33	2	14	13	38	6	0

Fig. 8. Holistic [22] and our proposed approach are compared among five major fine-grained action groups (i.e.,"cut","put in/on","take & put in","take out","open/close") in terms of per-class classification accuracy (true positive out of total)

There are still two major issues for our approach. First of all, object detection performance is far from good enough. For example, the object-of-interest list is coarse and incomplete, some defined object categories are difficult to detect (e.g., small size objects such as knife or vegetables, we group them as one object type in our work). Secondly, motions including human body or background motions (i.e., with mainly useless patterns) still count for a large part of dense trajectories, thus actions such as "cut" or "put" are easily confused by the intensive noise.

In the future work, we will make the localized semantic feature more discriminative and less noisy, e.g., by using better object detection method. Note that according to the large deformation and small size nature of the manipulated objects, superpixel based object detection is more suitable than DPM [9] in our scenario. But we believe the performance can be further improved if better tuned object detection method is applied. We can also leverage object co-occurrence information in the localized semantic feature extraction.

4 Conclusion

In summary, we propose a fine-grained action recognition pipeline which seamlessly incorporates localized semantic information into every processing step. The pipeline includes localized semantic feature extraction, semantic-grouped feature encoding, semantic-aware motion feature pooling and classification. We evaluate our approach on the MPII cooking fine-grained action dataset and achieve significant improvement over the existing methods, which is quite promising to be applied in applications such as daily living assist or medical assistance.

Acknowledgment. The study is supported by a research grant for the Human Sixth Sense Programme 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) 61128007, 2012 UTSA START-R Research Award.

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: Slic superpixels. EPFL 149300 (2010)
2. Bach, F.R., Lanckriet, G.R., Jordan, M.I.: Multiple kernel learning, conic duality, and the smo algorithm. In: ICML, pp. 6–13 (2004)
3. Cao, L., Mu, Y., Natsev, A., Chang, S.-F., Hua, G., Smith, J.R.: Scene aligned pooling for complex video recognition. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part II. LNCS, vol. 7573, pp. 688–701. Springer, Heidelberg (2012)
4. Chang, C.-C., Lin, C.-J.: Libsvm: a library for support vector machines. TIST 2(3), 1–27 (2011)
5. Chao, Y.-W., Yeh, Y.-R., Chen, Y.-W., Lee, Y.-J., Wang, Y.-C.F.: Locality-constrained group sparse representation for robust face recognition. In: ICIP, pp. 761–764 (2011)
6. Choi, W., Shahid, K., Savarese, S.: Learning context for collective activity recognition. In: CVPR, pp. 3273–3280 (2011)
7. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, pp. 886–893 (2005)
8. Fei-Fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: CVPR, pp. 524–531 (2005)
9. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. T-PAMI 32(9), 1627–1645 (2010)
10. Koppula, H.S., Gupta, R., Saxena, A.: Learning human activities and object affordances from rgb-d videos. CoRR (2012)
11. Lan, T.: Beyond actions: Discriminative models for contextual group activities. Ph.D. thesis, Applied Science: School of Computing Science (2010)
12. Lan, T., Wang, Y., Mori, G., Robinovitch, S.N.: Retrieving actions in group contexts. In: Kutulakos, K.N. (ed.) ECCV 2010 Workshops, Part I. LNCS, vol. 6553, pp. 181–194. Springer, Heidelberg (2012)
13. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR, pp. 1–8 (2008)
14. Lee, H., Battle, A., Raina, R., Ng, A.: Efficient sparse coding algorithms. In: NIPS, pp. 801–808 (2006)
15. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos "in the wild". In: CVPR, pp. 1996–2003 (2009)
16. Marszalek, M., Laptev, I., Schmid, C.: Actions in context. In: CVPR, pp. 2929–2936 (2009)
17. Moore, D., Essa, I., Hayes, M.: Exploiting human actions and object context for recognition tasks. In: ICCV, Greece (1999)
18. Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 143–156. Springer, Heidelberg (2010)
19. Rakotomamonjy, A., Bach, F.R., Canu, S., Grandvalet, Y.: Simplemkl. JMLR 9(11), 2491–2521 (2008)
20. Rohrbach, M., Amin, S., Andriluka, M., Schiele, B.: A database for fine grained activity detection of cooking activities. In: CVPR, pp. 1194–1201 (2012)
21. Ullah, M.M., Parizi, S.N., Laptev, I.: Improving bag-of-features action recognition with non-local cues. In: BMVC, vol. 10, pp. 1–11 (2010)

22. Wang, H., Klaser, A., Schmid, C., Liu, C.L.: Action recognition by dense trajectories. In: CVPR, pp. 3169–3176 (2011)
23. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: CVPR, pp. 3360–3367 (2010)
24. Wang, Y., Mori, G.: Hidden part models for human action recognition: Probabilistic versus max margin. T-PAMI 33(7), 1310–1323 (2011)
25. Willems, G., Tuytelaars, T., Van Gool, L.: An efficient dense and scale-invariant spatio-temporal interest point detector. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 650–663. Springer, Heidelberg (2008)
26. Wu, J., Osuntogun, A., Choudhury, T., Philipose, M., Rehg, J.: A scalable approach to activity recognition based on object use. In: ICCV, pp. 1–8 (2007)
27. Yao, B., Jiang, X., Khosla, A., Lin, A.L., Guibas, L., Fei-Fei, L.: Human action recognition by learning bases of action attributes and parts. In: ICCV (2011)
28. Yao, B., Khosla, A., Fei-Fei, L.: Classifying actions and measuring action similarity by modeling the mutual context of objects and human poses. In: ICML (2011)