LETTER

Modeling Interactions between Low-Level and High-Level Features for Human Action Recognition

Wen ZHOU[†a)], Chunheng WANG[†], Baihua XIAO[†], Zhong ZHANG[†], *Nonmembers*, *and* Yunxue SHAO[†], *Member*

SUMMARY Recognizing human action in complex scenes is a challenging problem in computer vision. Some action-unrelated concepts, such as camera position features, could significantly affect the appearance of local spatio-temporal features, and therefore the performance of low-level features based methods degrades. In this letter, we define the action-unrelated concept: the position of camera as high-level features. We observe that they can serve as a prior to local spatio-temporal features for human action recognition. We encode this prior by modeling interactions between spatio-temporal features and camera position features. We infer camera position features from local spatio-temporal features via these interactions. The parameters of this model are estimated by a new max-margin algorithm. We evaluate the proposed method on KTH, IXMAS and Youtube actions datasets. Experimental results show the effectiveness of the proposed method.

key words: *action recognition, camera position features, local spatio-temporal features, interactions*

1. Introduction

In the human-action recognition field, low-level features have evolved considerably. Sparse local spatio-temporal (ST) features have shown advantages in human action recognition [1], [2]. Besides these hand-designed features, hierarchical invariant ST features [3] that are learned from data directly are also proposed for human action recognition. They achieve impressive performance on many realistic video datasets. However, ST features are limited in capturing semantics information and often yield a representation with inadequate discriminative power for larger, more complex datasets. Consequently, high-level features are introduced to enrich ST features with semantic information and give a robust representation for human action recognition [4], [5].

Both low-level and high-level features have been explored considerably for human action recognition. However, little work has been developed for modeling interactions between low-level and high-level features. Here, action-unrelated concepts such as camera position features are defined as high-level features in the proposed method. We believe that there are strong connections between ST features and camera position features. Concretely, camera position features can be inferred from ST features via these connections. In addition, camera position features can be seen as a prior to guide ST features for human action recognition also via these connections. A more detailed description is shown in Fig. 1. In Fig. 1 (a), due to camera movement, a lot of unwanted background Spatio-Temporal Interest Points (STIPs) are detected from videos. However, only the discriminative STIPs reflect the camera position features, which are described in purple and green lines in Fig. 1 (a) respectively. Due to our discriminative learning, we build strong connections between discriminative features and the camera position features. Thus, camera position features can be inferred from local ST features.

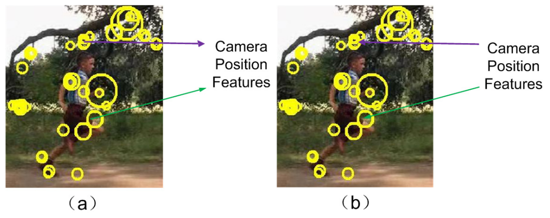

Fig. 1 Interactions between ST and camera position features. In (**a**), camera position features are inferred from ST features. In (**b**), camera position features are assigned as prior to ST features via connections between these two types of features.

In this letter, we model interactions between ST features and camera position features for human action recognition. Given a set of test videos, our model automatically infers the camera position and utilizes these features coupled with ST features to recognize human action simultaneously. The parameters of the model are estimated by a new max-margin algorithm. By modeling the interactions between ST features and camera position features, we show that our algorithm improves the performance of human action recognition on several action datasets.

2. Modeling Interactions between ST and Camera Position Features

Given a video, we recognize human action using ST features which are interacted with camera position features. The ST features extracted from these videos are suffered from a lot of noise due to the unconstraint environment. On the other hand, since videos contain human actions which are acted under different camera positions, ST features extracted from these videos are suffered from large variation.

A graphical illustration of our model is shown in Fig. 2. Since we do not know which ST feature is connected to specified camera position feature, we build a full connec-

Manuscript received March 4, 2013.
Manuscript revised July 26, 2013.

[†]The authors are with State Key Laboratory of Management and Control for Complex Systems Institute of Automation, Chinese Academy of Sciences, China.
a) E-mail: wen.zhou@ia.ac.cn

DOI: 10.1587/transinf.E96.D.2896

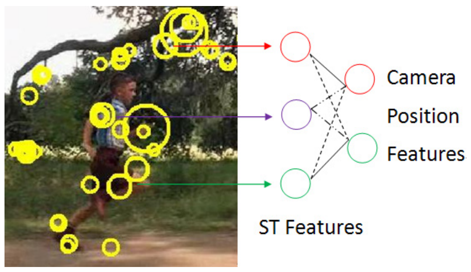

Fig. 2 A graphical illustration of our method. Our model can be thought as a shallow network, where camera position features interact with ST features directly.

tion between ST and camera features. Then, we formulate our model as follows:

$$y = \max_{\mathbf{h}\in\mathcal{H}}(\mathbf{x}^T\mathbf{w}\mathbf{h}), \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^D$ and $\mathbf{h} \in \mathcal{H}$ denote the ST and camera position features respectively, and $\mathbf{w} \in \mathbb{R}^{D\times H}$ is the weight of connections. Large values of $\mathbf{w}$ mean close relations between ST and camera position features.

The formulation of our model is similar with [6] in some respects. However, there are several significant differences between the proposed method and [6]. First, these two methods use different features which results in different inference and parameters learning algorithm. In the proposed method, $\mathbf{h}$ is unknown and we treat it as latent variable. In [6], Pirsiavash et al. only use low-level features (HOG features) to represent visual data. Second, since $\mathbf{h}$ is latent variable, the inference algorithm need to find the optimal value of $\mathbf{h}$ to maximize (1) and use the maximum value as the output of our classifier. However, Pirsiavash et al. predict the label by calculating the output of linear classifier in [6]. Third, in order to infer the latent variable $\mathbf{h}$ correctly and predict the class label accurately, we design our parameter learning algorithm to minimize two types of losses which are related to these two task while Pirsiavash et al. minimize the standard hinge loss to correctly predicts the class label.

3. Model Learning

Given the training videos of actions with labeled camera position features $\{\mathbf{h}_i\}_{i=1}^N$, where N is the number of training videos, the learning step needs to achieve two goals: to discover the camera position features and to estimate parameters for the potential weights to maximize the discrimination between different actions. In our model, these two goals are combined together, which is reflect by parameter $\mathbf{w}$. In other words, we discover camera position features using $\mathbf{w}$, and $\mathbf{w}$ also represents the parameters of discriminative model. To achieve these goals, we propose a novel max-margin learning framework for parameter estimation. Let $(\mathbf{x}_i, \mathbf{h}_i, y_i) \in \mathbb{R}^D \times \mathcal{H} \times \{-1, +1\}$ be a training sample, where $\mathbf{x}_i$ is ST feature, $\mathbf{h}_i$ is the camera position feature and y_i is the class label of $\mathbf{x}_i$. We use $\mathcal{P}$ and $\mathcal{N}$ to denote the positive and negative training set respectively.

We introduce two slack variables ξ_i and η_i for each sample $\mathbf{x}_i$ into objective function. These two slack variables have different purposes. The slack variable ξ_i denotes the hinge loss which penalizes the difference between the predicted class label and y_i. Minimizing the slack variable ξ_i aims at finding the parameters $\mathbf{w}$ to correctly predict the class label y_i. Meanwhile, minimizing the slack variable η_i aims at finding the parameters $\mathbf{w}$ to correctly predict the camera position features $\mathbf{h}_i$, which is obtained by maximizing the difference between $\mathbf{x}_i^T\mathbf{w}\mathbf{h}_i$ and $\max_{\mathbf{h}\in\mathcal{H}\setminus\mathbf{h}_i}\{\mathbf{x}_i^T\mathbf{w}\mathbf{h}\}$. This guarantees that the maximum score is obtained when $\mathbf{h}_i$ is correctly predicted. As for regularization term, since videos with different camera position features share little ST features, it is inappropriate to enforce other constraints on the parameters $\mathbf{w}$. Then, we only add the matrix norm regularization into objective function. Then, the objective function is defined as follows:

$$\begin{aligned}
&\min_{\mathbf{w},\xi,\eta} \frac{1}{2}\|\mathbf{w}\|_2^2 + \beta\sum_i \xi_i + \gamma\sum_i \eta_i \\
&s.t.\ \forall i,\ \xi_i \geq 0,\ \eta_i \geq 0 \\
&y_i(\mathbf{x}_i^T\mathbf{w}\mathbf{h}_i) \geq 1 - \xi_i \\
&\forall i \in \mathcal{P},\ \mathbf{x}_i^T\mathbf{w}\mathbf{h}_i \geq \max_{\mathbf{h}\in\mathcal{H}\setminus\mathbf{h}_i}\{\mathbf{x}_i^T\mathbf{w}\mathbf{h}\} + 1 - \eta_i \\
&\forall i \in \mathcal{N},\ \max_{\mathbf{h}\in\mathcal{H}}\{\mathbf{x}_i^T\mathbf{w}\mathbf{h}\} \leq -1 + \eta_i
\end{aligned} \tag{2}$$

where $\|\mathbf{w}\|_2$ is the L2 norm of $\mathbf{w}$, β and γ are normalization constants.

Since optimization problem (2) has a total of $N|\mathcal{H}| + |\mathcal{N}|$ constraints, it makes standard quadratic programming solvers unsuitable for this type of problem when $|\mathcal{H}|$ is extremely large. As in [7], we propose a learning algorithm which aims at finding a small set of active constraints that ensure a sufficiently accurate solution. It successively tighten the original problem using a cutting plane method. Pseudocode of the algorithm is depicted in Algorithm 1.

4. Model Inference

Define the set of action labels as $\mathcal{A} = \{1, \ldots, A\}$, we apply one-vs-all strategy for multi-class classification. We treat one action as positive examples and other actions as negative examples. Given a new test video V, histogram of ST features $\mathbf{x}$ is used to represent V. Then, inference is formulated as follows:

$$y = sgn(\max_{\mathbf{h}\in\mathcal{H}}(\mathbf{x}^T\mathbf{w}_k\mathbf{h})), \quad k = 1, \ldots, A. \tag{3}$$

It is achieved by finding a best configuration of camera position features to maximize the function (1).

Computational complexity: We use $|\mathcal{A}|$ to denote the total number of action class. Since for each action class, the inference process enumerates every possible configuration of camera position features, the total cost of inference is $|\mathcal{A}|D_H$, where D_H is the dimension of camera position features.

Algorithm 1: Algorithm for solving optimization problem (2)

```
   Input: $\epsilon$, $(\mathbf{x}_i, \mathbf{h}_i, y_i)$, $i = 1, \ldots, N$
1  Initialize: $S_i^{\mathcal{P}} \leftarrow \emptyset, S_i^{\mathcal{N}} \leftarrow \emptyset$,
2  $Q \leftarrow (\mathbf{x}_i, \mathbf{h}_i, y_i)$ $i = 1, \ldots, N$
3  repeat
4      for $i \in \mathcal{P}$ do
5          $Loss^{\mathcal{P}}(\mathbf{h}) = 1 - \mathbf{x}_i^T \mathbf{w}(\mathbf{h}_i - \mathbf{h})$,
6          where $\mathbf{w} \equiv \arg\min_{\mathbf{w}} \mathbf{h} \in S_j^{\mathcal{P}}$
7          compute $\hat{\mathbf{h}} = \arg\max_{\mathbf{h} \in \mathcal{H}} Loss^{\mathcal{P}}(\mathbf{h})$
8          compute $\eta_i = \max\{0, \max_{\mathbf{h} \in S_i^{\mathcal{P}}} Loss^{\mathcal{P}}(\mathbf{h})\}$
9          if $Loss^{\mathcal{P}}(\hat{\boldsymbol{h}}) > \eta_i + \epsilon$ then
10             $S_i^{\mathcal{P}} = S_i^{\mathcal{P}} \cup \{\hat{\mathbf{h}}\}$
11         end
12     end
13     for $i \in \mathcal{N}$ do
14         $Loss^{\mathcal{N}}(\mathbf{h}) = 1 + \mathbf{x}_i^T \mathbf{w}\mathbf{h}$,
15         compute $\hat{\mathbf{h}} = \arg\max_{\mathbf{h} \in \mathcal{H}} Loss^{\mathcal{N}}(\mathbf{h})$
16         compute $\eta_i = \max\{0, \max_{\mathbf{h} \in S_i^{\mathcal{N}}} Loss^{\mathcal{N}}(\mathbf{h})\}$
17         if $Loss^{\mathcal{N}}(\hat{\boldsymbol{h}}) > \eta_i + \epsilon$ then
18             $S_i^{\mathcal{N}} = S_i^{\mathcal{N}} \cup \{\hat{\mathbf{h}}\}$
19         end
20     end
21     $\mathbf{w} \leftarrow$ optimize object function (2) over $S \bigcup Q$,
       $S = S_i^{\mathcal{P}} \cup_i S_i^{\mathcal{N}}$
22 until no $S_i$ has changed during iteration;
```

5. Experiments

We adopt STIPs detector [8] to obtain local ST features. Bag-of-words (BOW) model is used to quantize these ST features. Then, a video is represented by histogram vector $\mathbf{x}$. In our experiment, we combine several sub-vocabularies to acquire the final vocabularies where sub-vocabularies are generated per action class separately.

KTH dataset. Considering the well-controlled environments of KTH action dataset, we define camera position feature $\mathbf{h}$ that it only involves the viewpoints. Specifically, $\mathbf{h}$ is a binary vector whose elements indicate the presence of specified viewpoints. We define $\mathbf{h}$ as 8-dimensional binary vector. $h_1, \ldots, h_8 \in \{0, 1\}$ denote that people act toward west, northwest, north, northeast, east, southeast, south and southwest respectively, where h_i $(i = 1, \ldots, 8)$ denotes the i-th element of $\mathbf{h}$. The size of sub-vocabularies are set to 1500.

IXMAS dataset. Due to the small moves of human body in this dataset, Harris3D detector fails to detect STIPs from some videos. Then, we employ primitive features in [9]. We perform a clustering of the feature space to quantize the primitive features. We represent activities using histograms of the frames assigned to each cluster. We use five dimensional vector to denote the camera position features. Each element of this vector denotes the presence of five different views in videos. Since our goal is not to recognize human action cross view, we do not use the common experimental setting. Instead, we apply 4-fold cross validation to demonstrate the advantage of the proposed model.

Youtube dataset. We obtain the average accuracy over 25-fold cross-validation. Since some videos in this dataset contain actions with different viewpoints, videos are segmented into several clips and each clip is labeled with one type of viewpoint. In this experiment, we only consider the variation of viewpoints. We define camera position features as 4 dimensional binary vector to capture the viewpoints' variation of human activities. These 4 binary variables indicate west, north, east, and south respectively. The size of sub-vocabularies is set to 5000. Then, the size of final vocabulary is 55000.

Baseline. In order to validate the advantages of our method, we use linear SVM coupled with BOW model as our baseline method. Liblinear tools [10] is utilized as classifier of baseline method.

5.1 Experimental Results

The camera position features for test samples are unknown. We obtain the camera position features automatically using (3) and also choose the class label using (3). The baseline method yields an average precision of 90.5%, whereas our model is 92.3% on KTH actions dataset. BOW model coupled with χ^2 kernel SVM [2] achieves 91.8%. Our method achieves 100%, 97.2%, 93.1%, 81.9%, 81.3% and 100% for boxing, hand clapping, hand waving, jogging, running and walking respectively, and the baseline method achieves 97.9%, 95.8%, 86.8%, 77.1%, 87.5% and 97.9% correspondingly. The main confusion occurs between jogging and running due to the similar appearance of these two actions. In most actions, our method achieves superior results compared with [2]. In order to validate the advantages of our model, we conduct a synthetic experiment which injects noise into BOW representation on KTH actions dataset. The noise is injected into bag-of-words representation for training data as follows: $\tilde{\mathbf{x}} = \mathbf{x} + R \cdot mnrnd(N^s, \mathbf{P}, 1)$, where $\mathbf{x}$ is BOW representation and $mnrnd(\cdot)$ is matlab function which generates random variable from $\mathbf{P}$. $\mathbf{P}$ is a multinomial distribution whose elements $p_k = \frac{1}{d}$, $k = 1, \ldots, d$, where d is the dimension of $\mathbf{x}$. R and N^s are two parameters which control the strength of injected noise. Higher values of R and N^s indicate more noise in $\tilde{\mathbf{x}}$. Figure 3 reports the performance of our method and the baseline method after injecting noise when K = 800. With the higher value of both R and N^s, our method significantly outperforms the baseline method which uses linear SVM. As shown in Fig. 3, with increasing values of R and N^s, the recognition accuracy of our method almost does not degrade, while the linear SVM degrades significantly. This supports our claim that the proposed method discovers discriminative co-occurrence information for ST and camera position features, which is robust to noise.

Since Sadanand and Corso [5] use richer representation for videos, they report a recognition accuracy of 98% on KTH actions dataset. However, there are three main advantages of our method compared to [5]. First, our method automatically reports the high-level features (camera position features) and provides a semantic description for videos. Second, since the method described in [5] uses FFT-based

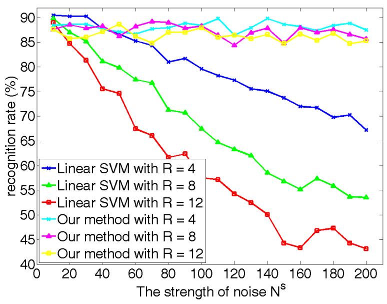

Fig. 3 Performance of our method and linear SVM after noise injecting on KTH action dataset. The blue, green and red lines denote the performance of linear SVM with $R = 4$, 8 and 12 respectively, and cyan, magenta, yellow lines denote the performance of our method with $R = 4$, 8 and 12 respectively.

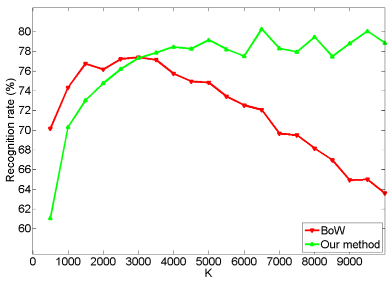

Fig. 4 Comparison of our method and our baseline with different sizes of the sub-vocabularies on IXMAS action dataset.

Table 1 Average accuracy on Youtube actions dataset by action class.

Actions	[11]	[3]	[12]	Baseline[1]	Baseline[2]	Ours
Cycle	73	86.9	-	72.2	88.2	**93.9**
Dive	81	93	-	88	97	**99**
Golf	86	85	-	95	**98**	**98**
Juggle	54	64	-	72	90	**89**
Jump	79	87	-	81	89	**99**
Ride horse	72	76	-	67	79	**84**
Basketball	53	46.5	-	81	87	**90**
Volleyball	73.3	81	-	94	**100**	**100**
Swing	57	88	-	76	87	**95**
Tennis	80	56	-	90	96	**97**
Walk	75	78.1	-	55.9	**86.1**	82.3
Average	71.2	76.5	87.0	79.3	91.5	**93.7**

convolution, it is computational inefficient compared to our method. Third, our method is a framework to incorporate different types of features. Other combinations of different type features can also applied into our framework including local ST features and action-related concepts such as the movements of body parts (*e.g.* legs, hands and knee).

The best result of our method achieves 80.3% on IXMAS action dataset while BOW model achieves 77.4%, shown in Fig. 4. Figure 4 gives the comparison of our method and the baseline with different sizes of the sub-vocabularies. From Fig. 4 we can see that, our baseline achieves the best result with K = 3000, and our method is 6500, where K is the size of sub-vocabularies.

Table 1 gives the average accuracy and comparison with other methods on Youtube actions dataset by action class. Baseline[1] and Baseline[2] denote the baseline method with K = 800, and K = 5000 respectively. Due to the large size of final vocabulary (5000*11 class), our baseline method achieves 91.5%. Our baseline method achieves 82.8%, 84.4%, 87.8%, 90.8% and 91.5% when K = 1000, 2000, 3000, 4000 and 5000 respectively. We achieve 93.7% recognition rate for the Youtube actions by modeling interactions between camera position features and spatio-temporal features. It is approximately 7% better than the current best published results. Our method obtains better results compared to [3], [11]. Our method also gives better results compared to our baseline method since our camera position features capture the camera position variation for this complex and challenging dataset.

6. Conclusion

In this work, we model interactions between low-level and high-level features for human action recognition. We observe that camera position features (high-level features) can be inferred from ST features (low-level features), and camera position features can serve as prior to ST features. By modeling the interactions, our method can infer camera position features from ST features automatically and also imposes constraints on ST features. We evaluate our method on several action datasets to demonstrate the advantages of our method. Although we focus on the interactions between local ST features and camera position features in this work, other combinations of different type features can also applied into our framework.

References

[1] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," IEEE International Workshop on VSPETS, pp.65–72, 2005.

[2] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," CVPR, pp.1–8, 2008.

[3] Q.V. Le, W.Y. Zou, S.Y. Yeung, and A.Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," CVPR, 2011.

[4] J. Liu, B. Kuipers, and S. Savarese, "Recognizing human actions by attributes," CVPR, pp.3337–3344, 2011.

[5] S. Sadanand and J. Corso, "Action bank: A high-level representation of activity in video," CVPR, pp.1234–1241, 2012.

[6] H. Pirsiavash, D. Ramanan, and C. Fowlkes, "Bilinear classifiers for visual recognition," NIPS, vol.22, pp.1482–1490, 2009.

[7] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, "Support vector machine learning for interdependent and structured output spaces," ICML, p.104, 2004.

[8] I. Laptev, "On space-time interest points," IJCV, vol.64, no.2, pp.107–123, 2005.

[9] A. Farhadi and M. Tabrizi, "Learning to recognize activities from the wrong view point," ECCV, pp.154–166, 2008.

[10] R.E. Fan, K.W. Chang, C.J. Hsieh, X.R. Wang, and C.J. Lin, "Liblinear: A library for large linear classification," J. Machine Learning Research, vol.9, pp.1871–1874, 2008.

[11] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos in the wild," CVPR, pp.1996–2003, 2009.

[12] B. Chakraborty, M. Holte, T. Moeslund, J. Gonzalez, and F. Roca, "A selective spatio-temporal interest point detector for human action recognition in complex scenes," ICCV, pp.1776–1783, 2011.