A Joint Evaluation of Dictionary Learning and Feature Encoding for Action Recognition

Xiaojiang Peng[1,3], Limin Wang[2,3], Yu Qiao[3], Qiang Peng[1]
[1]School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China
[2]Department of Information Engineering, The Chinese University of Hong Kong
[3]Shenzhen key lab of Comp. Vis. & Pat. Rec., Shenzhen Institutes of Advanced Technology, CAS, China
{xiaojiangp, 07wanglimin}@gmail.com, yu.qiao@siat.ac.cn, qpeng@swjtu.edu.cn

***Abstract*—Many mid-level representations have been developed to replace traditional bag-of-words model (VQ+k-means) such as sparse coding, OMP-k with k-SVD, and fisher vector with GMM in image domain. These approaches can be split into a dictionary learning phase and a feature encoding phase which are often closely related. In this paper, we jointly evaluate the effect of these two phases for video-based action recognition. Specially, we compare several dictionary learning methods and feature encoding schemes through extensive experiments on the KTH and HMDB51 datasets. Experimental results indicate that fisher vector performs consistently better than the other encoding methods, and sparse coding is robust to different dictionaries even random weights. In addition, we observe that the advantages of sophisticated mid-level representations do not come from their specific dictionaries but the encoding mechanisms, and we can just use randomly selected exemplars as dictionaries for most of encoding methods. Finally, we achieve the state-of-the-art results on the HMDB51 and UCF101 by combining our configurations with improved dense trajectory features.**

I. Introduction

Human action recognition has been an active research area in recent years due to its wide applications, such as smart video surveillance, video indexing and human-computer interface [1]–[9]. The difficulty of action recognition comes from the large intra-class variation, clutter and occlusion in background or foreground and other fundamental difficulties. In recent years, much work has applied bag-of-words (BoW) model [10] in human action recognition [1]–[4]. This model mainly contains five steps: feature extraction, dictionary learning, feature encoding, pooling, and normalization. As for classical BoW, we usually extract local features from videos, learn a visual dictionary in training set by clustering algorithm like k-means, encode local features to their nearest words (i.e., vector quantization (VQ)), and finally create a histogram for each video by aggregating the frequency of visual words.

Many recent efforts have been devoted to the dictionary learning and feature encoding phases for visual recognition due to their importance. Aharon *et al*. [11] presented a singular value decomposition (k-SVD) based approach to learn effective over-completed dictionary which is a generalized version of k-means. Sometimes it is also known as OMP-k since the orthogonal matching pursuit (OMP) is utilized to assign features, which is an approximate solution for ℓ_0 norm sparse representation. Lee *et al.* [12] developed a ℓ_1 norm based sparse coding (SC) algorithm, where feature-sign search algorithm was applied for encoding and Lagrange dual method for dictionary learning. Yang *et al.* [13] employed this SC scheme for image classification and achieved excellent performance. Wang *et al.* [14] proposed a locality-constrained linear coding (LLC) where a locality constraint is added to the loss function of SC. Liu *et al.* [15] presented local soft-assignment (SA-k) for image classification. Perronnin *et al*. [16] developed an improved fisher vector (FV) with dictionary of Gaussian mixture model (GMM) for object recognition. Wang *et al*. [17] utilized FV for action recognition. It is worth noting that each encoding method is equipped with its specific dictionary learning algorithm. For example, dictionary learning method of k-SVD for encoding method OMP-k, SC for SC, and GMM for FV. A natural question is "*what is the correlation between dictionary learning method and encoding method, and which is more important for performance improvement?*".

In this paper, we investigate the mutual influence of dictionary learning methods and feature encoding approaches for action recognition in video domain. As for video-based action recognition, there exist several evaluations for the local spatial-temporal feature extraction [4], feature encoding, pooling and normalizing methods [17]. To the best of our knowledge, there is still no reported work on the joint evaluation of dictionary learning and feature encoding methods in the context of human action recognition. Specially, the selected dictionary learning approaches in this paper are namely random weights (RW) which selects random numbers yielded by uniform distribution as dictionary, random exemplars (RE) which randomly collects features from training set as dictionary, k-means, GMM, k-SVD (or OMP-k) [11] and sparse coding [12]. To investigate the performance with different encoding schemes, we employ several feature encoding methods, namely VQ, soft-assignment [15], OMP, SC, LLC [14], and fisher vector [16].

The main contributions of this paper come from our observations: (1) it's not necessary to keep specific dictionary learning methods for certain encoding methods; (2) encoding methods play the leading role for performance improvement; (3) fisher vector performs consistently better than the other encoding methods for human action recognition; (4) sparse coding is robust to different dictionaries even random weights. Finally, we achieve the state-of-the-art performance on two large datasets–HMDB51 [5] and UCF101 [6].

II. Method Review

The video representation of our evaluation model is shown in Figure 1. First, local spatial-temporal features are extracted (e.g., STIP [3], Cuboids [2], and improved dense trajectories [7]), and then a dictionary is learned from the features in

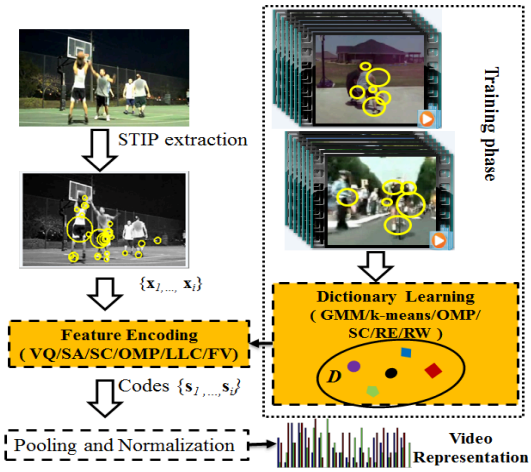


Fig. 1. Video representation of our evaluation framework.

training set. Next, features are encoded using the learned dictionary, and then all the coding coefficients in a single video are pooled and normalized as the final video representation. Here, we focus on the dictionary learning and feature encoding steps. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N] \in \mathcal{R}^{d\times N}$ be a set of local features, $\mathbf{D} \in \mathcal{R}^{d\times K}$ be the learned dictionary and $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_N] \in \mathcal{R}^{K\times N}$ be the coefficient vectors. Following are the details of different dictionary learning and feature encoding methods.

A. Dictionary Learning Methods

Dictionary learning aims to obtain a certain dictionary $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_K] \in \mathcal{R}^{d\times K}$, which can best depict the input feature space. Here, we give the formulations of those methods we used.

Random Weights (RW). Following [18], we obtain a dictionary by filling the columns of $\mathbf{D}$ with vectors sampled from a unit normal distribution (subsequently normalized to unit length).

Randomly selected Exemplars (RE). This approach just fills the columns of $\mathbf{D}$ with normalized vectors randomly sampled from training data set $\mathbf{X}$.

k-means. It is perhaps the most popular unsupervised way to learn a dictionary due to its simplicity and effectiveness. It aims to minimize the following objective function:

$$C = \sum_{i=1}^{K}\sum_{j=1}^{n_i} ||\mathbf{x}_j^{(i)} - \mathbf{d}_i||_2^2, \tag{1}$$

where $\mathbf{x}_j^{(i)}$ denotes the data point included in the cluster i whose center is $\mathbf{d}_i$. In practice, we first select K data points as initial centroids, then assign each data point to the closest centroid and recalculate the positions of the K centroids, repeat these until the centroids no longer move.

GMM. Gaussian Mixture Model is a probabilistic model to depict the distribution over the given feature space:

$$p(\mathbf{x};\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x};\mu_k,\Sigma_k), \tag{2}$$

where K is the number of mixture components and $\theta = \{\pi_1, \mu_1, \Sigma_1, \cdots, \pi_K, \mu_K, \Sigma_K\}$ is the model parameters. $\mathcal{N}(\mathbf{x};\mu_k,\Sigma_k)$ is a d-dimensional Gaussian distribution. The optimal parameters of GMM can be learned through maximum likelihood $\ln p(\mathbf{X};\theta) = \sum_n \ln p(x_n;\theta)$ by employing the iterative EM algorithm [19].

k-SVD (**OMP-k**). As a generalized version of k-means, **k-SVD** [11] has an alternative objective function:

$$\min_{\mathbf{D},\mathbf{s}_i} \sum_{i=1}^{N} ||\mathbf{x}_i - \mathbf{D}\mathbf{s}_i||_2^2, \tag{3}$$
$$\text{s.t.} \quad \forall i, ||\mathbf{s}_i||_{\ell_0} \le k \quad \text{and} \quad \forall j, ||\mathbf{d}_j||_2^2 = 1,$$

where k is the largest number of non-zero components in each code $\mathbf{s}_i$. Usually, the codes are computed using OMP algorithm, thus we refer it to as OMP-k here in order to keep pace with its corresponding encoding method. For each single input $\mathbf{x}_i$, OMP greedily selects the most relevant $\mathbf{d}_i$ at each iteration and makes an element of $\mathbf{s}_i$ to be non-zero. After the k-th selection, $\mathbf{s}_i$ is updated to minimize $||\mathbf{x}_i - \mathbf{D}\mathbf{s}_i||_2^2$ with allowing only the selected elements to be non-zero. And after computing all the codes, we update the elements of $\mathbf{D}$ one by one via applying SVD to the residual [11].

Sparse Coding (SC). Here, sparse coding dictionary specially denotes the ℓ_1-norm constrained one from Lee *et al.* [12]. The standard objective function of ℓ_1-norm constrained sparse coding based dictionary learning is as follows,

$$\min_{\mathbf{D},\mathbf{s}_i} \sum_{i=1}^{N} ||\mathbf{x}_i - \mathbf{D}\mathbf{s}_i||_2^2 + \lambda \sum_i ||\mathbf{s}_i||_{\ell_1}, \quad \text{s.t.} \forall j, ||\mathbf{d}_j||_2^2 = 1, \tag{4}$$

where λ is a sparse factor. Lee et al. [12] proposed a feature-sign search algorithm to solve the ℓ_1-regularized least squares (i.e., encoding) problem and a Lagrange dual technique to work out the ℓ_2-constrained least squares (i.e., dictionary updating). The dictionary is generated iteratively. The main idea of feature-sign search algorithm is to guess the signs of coding coefficients from "helpful" features and then solve an unconstrained quadratic optimization problem.

B. Feature Encoding Methods

The purpose of feature encoding is to compute a vector $\mathbf{s} \in \mathcal{R}^K$ for input $\mathbf{x}$ with $\mathbf{D}$. Here, we give the formulations of the encoding methods we used.

Vector quantization (VQ). VQ is the standard encoding method of BoW, which solves the following constrained objective function:

$$\mathbf{s}(i) = 1, \text{if} \quad i = \arg\min_j ||\mathbf{x} - \mathbf{d}_j||_2^2, s.t. ||\mathbf{s}||_{\ell_0} = 1 \tag{5}$$

where constraint $||\mathbf{s}||_{\ell_0} = 1$ means that there will be only one non-zero element in $\mathbf{s}$, which is found by searching the nearest word in the dictionary.

Soft-assignment (SA). SA means that more than one word will be used. In fact, there are several techniques to realize soft-assignment (e.g., [15], [20], [21]). We select the k-nearest neighborhood or "localized" version of Liu's [15]

(here we name it as SA-k) in our experiments due to its good performance. The k elements of vector $\mathbf{s}$ are given by,

$$\mathbf{s}(i) = \begin{cases} \frac{\exp(-\beta||\mathbf{x}-\mathbf{d}_i||_2^2)}{\sum_{i=1}^{k}\exp(-\beta||\mathbf{x}-\mathbf{d}_i||_2^2)}; & \text{if } \mathbf{d}_i \in N_k(\mathbf{x}), \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

Where $N_k(\mathbf{x})$ denotes the k-nearest neighborhood of $\mathbf{x}$. β is a smoothing factor to control the softness of the assignment.

Sparse coding (SC). Given a dictionary $\mathbf{D}$, SC tries to get the code $\mathbf{s}$ for input $\mathbf{x}$ by solving the following function:

$$\mathbf{s} = \arg\min_{\mathbf{s}} ||\mathbf{x} - \mathbf{D}\mathbf{s}||^2 + \lambda||\mathbf{s}||_{\ell_1}, \quad (7)$$

This problem is well known as the *lasso* problem [22]. Several algorithms can be used to solve this problem such as least angle regression [23] and feature-sign method [12]. We employ feature-sign scheme here.

Orthogonal Matching Pursuit (OMP-k). As mentioned above, given $\mathbf{x}$ and $\mathbf{D}$, we greedily select the most relevant $\mathbf{d}_i$ at each iteration and make an element of $\mathbf{s}$ to be non-zero. After the k-th selection, $\mathbf{s}$ is updated to minimize $||\mathbf{x} - \mathbf{D}\mathbf{s}||_2^2$ by allowing only the selected elements to be non-zero.

Locality-constrained linear coding (LLC). Wang [14] suggested that locality is more essential than sparsity, since locality must lead to sparsity but not necessary vice versa. The coefficient vector of LLC is obtained by solving the following optimization:

$$\mathbf{s} = \arg\min_{\mathbf{s}} ||\mathbf{x} - \mathbf{D}\mathbf{s}||^2 + \lambda||\mathbf{e} \odot \mathbf{s}||^2, \qquad \text{s.t.} \quad \mathbf{1}^\top\mathbf{s} = 1, \quad (8)$$

where $\mathbf{e} = \exp(\text{dist}(\mathbf{x}, \mathbf{D})/\sigma)$ and $\text{dist}(\mathbf{x}, \mathbf{D})$ denotes the Euclidean distance between $\mathbf{x}$ and $\mathbf{D}$. σ is a parameter controlling the weight vector $\mathbf{e}$. In our experiments, we apply the k-NN version of LLC (here we call it LLC-k), which is an approximation with low computational cost in practice.

Fisher Vector (FV). Fisher vector is derived from fisher kernel which is introduced for large-scale image categorization [16]. The fisher kernel is a generic framework which combines the benefits of generative and discriminative approaches. It usually leverages GMM as its dictionary. As it is known, the gradient of the log-likelihood with respect to a parameter can describe how that parameter contributes to the process of generating a particular example. Then the video can be described by the gradient vector of log likelihood with respect to the model parameters [24]:

$$G_\theta^{\mathbf{X}} = \frac{1}{N}\nabla_\theta \log p(\mathbf{X}; \theta). \quad (9)$$

Note that the dimensionality of this vector depends only on the number of parameters in θ. Perronnin *et al*. [16] developed an improved fisher vector as follows,

$$\mathcal{G}_{\mu,k}^{\mathbf{X}} = \frac{1}{N\sqrt{\pi_k}}\sum_{n=1}^{N}\gamma_n(k)\left(\frac{\mathbf{x}_n - \mu_k}{\sigma_k}\right), \quad (10)$$

$$\mathcal{G}_{\sigma,k}^{\mathbf{X}} = \frac{1}{N\sqrt{2\pi_k}}\sum_{n=1}^{N}\gamma_n(k)\left[\frac{(\mathbf{x}_n - \mu_k)^2}{\sigma_k^2} - 1\right], \quad (11)$$

where $\gamma_n(k)$ is the weight of local feature $\mathbf{x}_n$ to i-th Gaussian:

$$\gamma_n(k) = \frac{\pi_k\mathcal{N}(\mathbf{x}_n; \mu_k, \Sigma_k)}{\sum_{i=1}^{K}\pi_i\mathcal{N}(\mathbf{x}_n; \mu_i, \Sigma_i)}. \quad (12)$$

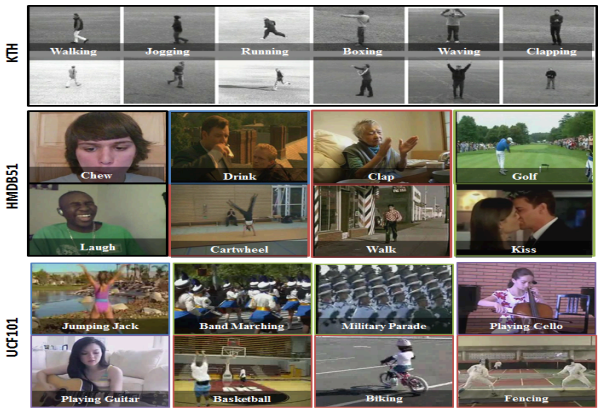


Fig. 2. Sample frames from KTH, HMDB51, and UCF101 datasets.

The final fisher vector is the concatenation of $\mathcal{G}_{\mu,k}^{\mathbf{X}}$ and $\mathcal{G}_{\sigma,k}^{\mathbf{X}}$ which is a $2Kd$ dimensional super vector.

III. Experimental Evaluation and Discussion

A. Datasets

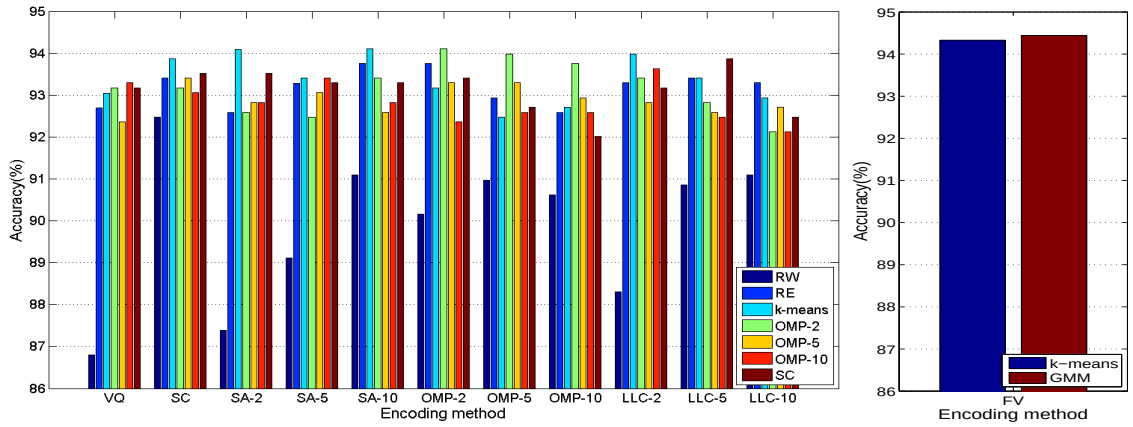
We conduct experiments on three published action datasets–KTH [1], HMDB51 [5], and UCF101 [6]. These datasets are collected from controlled experimental setting and web videos. Totally, we use more than 22,000 video clips. Some sample frames are illustrated in Figure 2.

The **KTH** dataset [1] is one of the most popular datasets in action recognition, which consists of 2,391 video clips acted by 25 subjects. It contains 6 action classes: *walking*, *jogging*, *running*, *boxing*, *hand-waving*, and *hand-clapping*. Actions are recorded at 4 environment settings: outdoors, outdoors with camera motion, outdoors with clothing change, and indoors. We follow the experimental settings in [1] where clips are divided into a training set (16 subjects) and a testing set (9 subjects). The **HMDB51** dataset [5] consists 51 action categories with 6,766 manually annotated clips which are extracted from a variety of sources ranging from digitized movies to YouTube. We follow the experimental settings in [5] where three train/test splits are available, and report the mean average accuracy over all classes. The **UCF101** dataset [6] is perhaps the largest action recognition dataset currently and gives the largest diversity in terms of actions, with the presence of large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions, etc. It contains 13,320 videos collected from YouTube and includes total number of 101 action classes. The videos are grouped into 25 groups. We perform evaluation on the newest three train/test splits [1] and report the mean average accuracy over all classes.

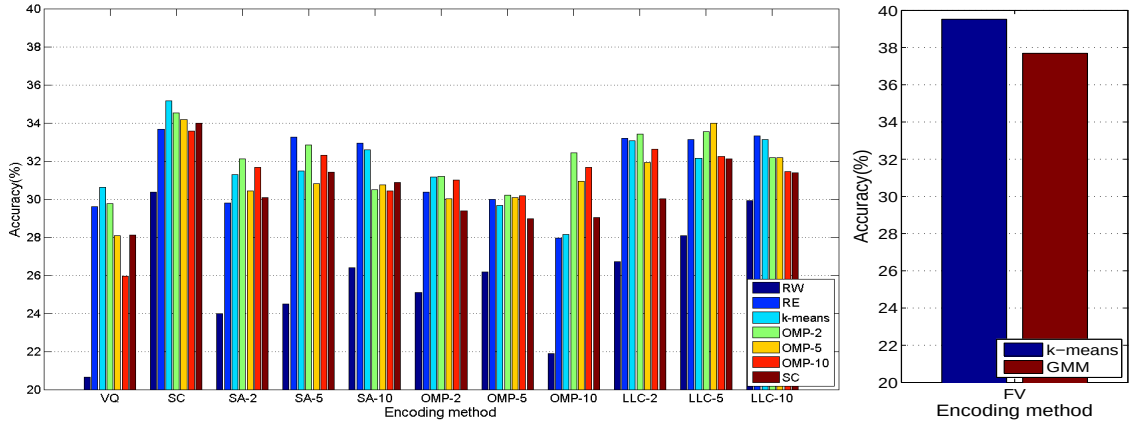
B. Joint exploration of dictionary learning algorithms and encoding methods

We firstly conduct expansive experiments on the KTH and HMDB51 datasets with different dictionary learning algorithms and encoding methods.

[1] http://crcv.ucf.edu/ICCV13-Action-Workshop/



(a) KTH



(b) HMDB51

Fig. 3. Results of different dictionaries and encoding schemes on the KTH and HMDB51 datasets. Note there are only k-means and GMM for fisher vector because other dictionary learning methods can not provide the necessary parameters for FV.

Implementation details. Spatial-temporal interest points (STIPs) are extracted for both datasets using the version 1.1 of source code from the author's website [3]. We separate HOG and HOF descriptors, and construct two BoW models. Specifically, we randomly sample 100k features to learn dictionaries with size of 4k. When the FV is used, the dictionary size (GMM mixture number) is fixed to 256. In order to evaluate the effect of the number of neighbours for those locality based encoding methods, we choose $k = [2, 5, 10]$ for OMP-k, SA-k and LLC-k, and set the mentioned parameters $[\beta, \lambda, \sigma]$ to be [1, 0.15, 1]. As for dictionary learning phase, we use the VL_Feat toolbox for k-means, employ the source code from Yang's website [13] for sparse coding, and set 50 iterations for both OMP and SC. After encoding all the features, we employ sum pooling for all the methods and normalized by "SSR+L2" scheme which demonstrated better results than other post-processing strategies by our previous work [17]. We compute RBF kernels with χ^2 distance for both HOG and HOF channels [3] for all the encoding methods except for FV, and then get the average kernel as inputs for kernel SVM classifier. When evaluating GMM and k-means for FV (the parameters for FV are directly computed from all the clusters when using k-means), we perform PCA and whitening for both HOG and HOF with dimensions of 40 and 60, and the final vectors are directly concatenated and sequently input a linear SVM. As for multi-class classification, we use the *one-against-rest* approach and select the class with the highest score.

Key observations and analysis. We explore the possible combinations of different dictionary learning and feature encoding methods, and the results are shown in Figure 3. Several observations can be obtained from Figure 3. *Firstly*, there is no evidence that a certain encoding method should utilize its specific dictionary learning algorithm. For instance, the result of "k-means+FV" is very similar to that of "GMM+FV" on KTH and even better than on HMDB51. *Secondly*, all the dictionary learning methods except RW can be adapted to different feature encoding algorithms and obtain the similar performance on both the datasets. We analyze that RW sampling uniformly in the feature space, therefore fail to capture the heavy tail property of local feature distribution. The other dictionary learning methods consider the structure of feature space either implicitly or explicitly. *Thirdly*, comparing different encoding methods, we conclude that FV obtains the best performance and VQ obtain the worst performance. This result can be ascribed to the fact that FV reserves the richest information, i.e. the first-order and second-order statistics, during the encoding process, where VQ just store the occurrence of words, i.e. zero-order statistics. For other encoding methods, SC performs slightly better and the rest obtain similar results. These methods all leverage the "sparsity" into their framework essentially and capture similar aspect of feature space. *Finally*, we notice

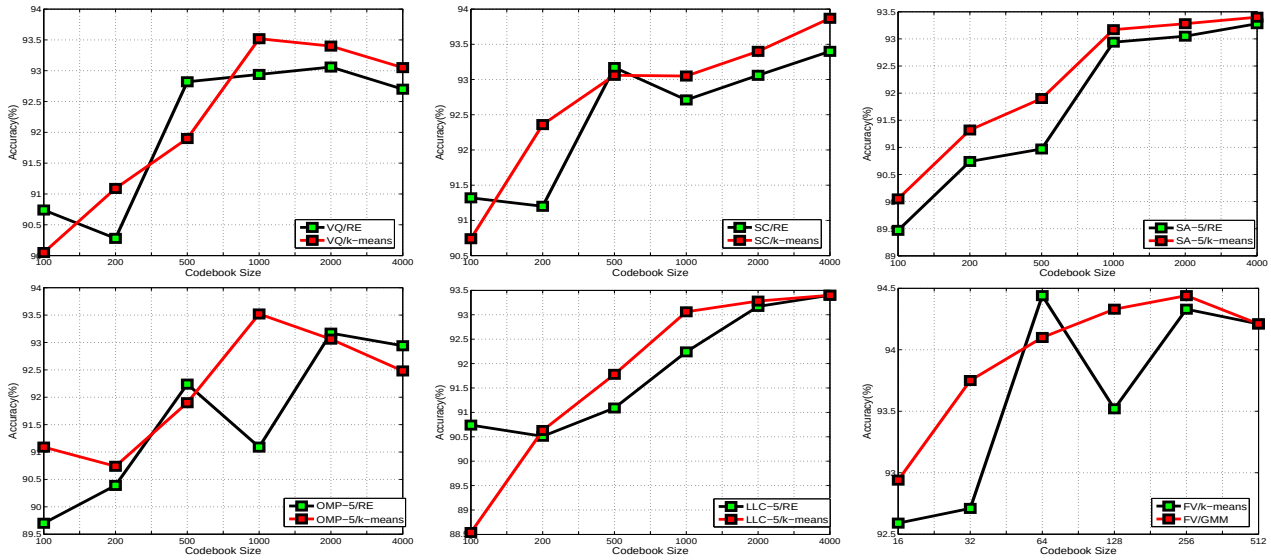


Fig. 4. Performance of different encoding methods using k-means, RE, and GMM with changing dictionary sizes on the KTH dataset.

that sparse coding is robust to different dictionaries, and OMP, LLC, SA is not sensitive to the number of nearest neighbours.

Dataset complexity. All of the methods obtain similar results (except for the methods with RW) on KTH but there are relatively large differences on HMDB51. The classical "k-means+VQ" and "GMM+FV" achieve 93.29% and 94.44% on KTH, respectively. But they obtain 29.6% and 37.69% on HMDB51, respectively. The distinctions between them come from the complexity of action videos and the number of categories. The KTH dataset owns only 6 classes and the videos are recorded in a constrained environment. The diversity of videos is much lower than that of the ones from HMDB51. The dictionary of size 4,000 is able to cover the complexity of feature space of the KTH, and the the histograms of different classes can be separated effectively, no matter which encoding method is used. Thus, all the approaches perform similarly on the KTH. However, for HMDB51, the number of classes is much larger and the STIPs are not only extracted in motion foreground but also in the background due to serious camera motion. Therefore, it is not enough to cover the feature space for each class with less than 80 words because of large variations, which results in large overlaps among the word distributions of different classes on the HMDB51 dataset.

Dictionary size. We further discuss the influence of dictionary size on the KTH dataset. We take RE, k-means, and GMM for this purpose. The results of several encoding methods with different dictionary sizes are shown in Figure 4. Generally, the results are improved with increasing dictionary size. The appropriate dictionary size is 1,000 for VQ and OMP-5, and 4,000 for SC, SA-5, and LLC-5 from Figure 4. The improvement by increasing the dictionary size with k-means is more reliable than RE but the global trend is similar, and the same case is for GMM versus k-means.

Cost. Table I shows the detailed costs of dictionary learning schemes. The time consumptions are ranked as: RE≈RW$\ll$ k-means$\ll$OMP-2$<$OMP-5$<$OMP-10$<$SC. RE and RW are the fastest ones obviously, and the others have to spend expensive time cost for optimization. Dictionary using SC almost takes one day which mainly due to the repeated encoding steps for those 100k features. We also compare the costs of different encoding methods by randomly selected 50 videos from the KTH dataset. The dictionaries are all generated by k-means with the size fixed to 4000 except for FV. The dictionary size for FV is set to 256. The average costs of encoding strategies for single video are illustrated in Figure 5. The runtime is obtained on an Acer laptop with a 2.5 GHz Intel Core i5 CPU and 4 GB RAM. And we set 50 iterations for OMP and SC.

TABLE I. THE COST OF DIFFERENT DICTIONARY LEARNING METHODS WITH THE SIZE OF 4K FROM 100K FEATURES ON KTH DATASET.

Methods	RW	RE	K-means	OMP-2	OMP-5	OMP-10	SC
HOG	0.15s	0.15s	7.48min	1.45h	2.83h	5.50h	23.14h
HOF	0.18s	0.18s	7.50min	1.75h	3.05h	5.94h	31.47h

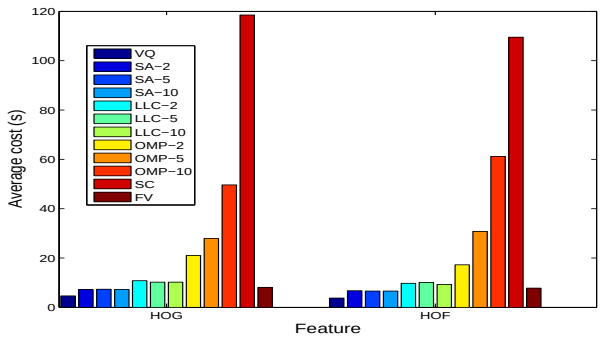


Fig. 5. The average cost of different feature encoding methods for a video on KTH, where the dictionary is fixed by k-means with the size of 4k.

C. Further exploration with improved dense trajectory feature

In this section, we further study these mid-level representations with improved dense trajectories (IDTs) on the HMDB51 and UCF101 dataset due to their good performance [7]. IDTs extract four kinds of descriptors, namely HOG, HOF, MBHx, and MBHy, and all the descriptors are preprocessed by PCA and whitening. Due to the observation that dictionary

TABLE II. THE RESULTS OF DIFFERENT ENCODING METHODS WITH IDT FEATURE ON HMDB51 AND UCF101 DATASETS.

	HMDB51				UCF101			
Methods	VQ	SA-5	LLC-5	FV	VQ	SA-5	LLC-5	FV
HOG	34.81	35.45	34.68	42.72	65.4	65.81	65.46	74.64
HOF	42.07	42.66	42.18	50.81	70.57	71.14	71.03	77.95
MBHx	34.6	35.51	35.51	44.23	66.43	67.55	67	74.76
MBHy	39.78	40.35	40.39	49	68.5	69.67	69.6	76.94
Combine	55.27	55.8	55.45	59.7	81.37	81.65	81.57	86.57

TABLE III. COMPARISON WITH THE STATE OF THE ART.

KTH		HMDB51		UCF101	
Methods	Acc.	Methods	Acc.	Methods	Acc.
[3]	91.8%	[17]	31.8%	Winner[1]	85.9%
[25]	95.6%	[26]	42.1%		
[2]	81.16%	[27]	46.6%		
[17]	92.1%	[7]	57.2%		
Ours	94.44%	Ours	**59.7**%	Ours	**86.57**%

learning have little influence on classification performance in previous section, we just explore the performance of different encoding methods with IDT. Table II shows the results of several encoding methods. The dictionaries are yielded by k-means with size of 8k for VQ, SA-5, and LLC-5, and 512 for FV. We do not evaluate SC due to that it is impractical to employ SC for encoding more than 1 billion features (the estimated cost is half a year by a laptop with dual-core).

The same observation can be found in Table II. Fisher vector performs best on both datasets as expected. Both SA-5 and LLC-5 obtain slight better results than VQ for single feature. However, we also notice that the performance gap among these encoding methods becomes smaller when multiple descriptors are fused. We explain that the complementary property among different descriptors reduce the influence of different encoding methods on final performance.

D. Comparison with the state of the art

Table III compares our best results with the state of the art. For the KTH dataset, we see our result is comparable to most of recently published results by just using STIP feature. For both the HMDB51 and UCF101 datasets, we achieve slight better results than the state-of-the-art results.

IV. CONCLUSION

In this paper, we conduct extensive experiments to evaluate the effects of dictionary learning methods and encoding schemes for human action recognition. We find that it is not necessary to keep specific dictionary learning methods for certain encoding methods. We also analyze the results from the views of dataset complexity, dictionary size, and the computational cost. Finally, we achieve the state-of-the-art results on the HMDB51 and UCF101 datasets by applying our best configuration with improved dense trajectory fearures.

Acknowledgments This work is partly supported by National Natural Science Foundation of China (91320101, 60972111), Shenzhen Basic Research Program (JC201005270350A, JCYJ 20120903092050890, JCYJ20120617114614438, JCYJ201 30402113127496), 100 Talents Program of CAS, and Guangdong Innovative Research Team Program (No. 2010 01D0104648280), and the 2013 Doctoral Innovation Funds of Southwest Jiaotong University. Yu Qiao is the corresponding author.

REFERENCES

[1] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local svm approach," in *ICPR*, 2004.

[2] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *PETS*, 2005, pp. 65–72.

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

[4] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, C. Schmid *et al.*, "Evaluation of local spatio-temporal features for action recognition," in *BMVC*, 2009.

[5] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: A large video database for human motion recognition," in *ICCV*.

[6] K. Soomro, A. Roshan Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," in *CRCV-TR-12-01*, 2012.

[7] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *ICCV*, 2013.

[8] L. Wang, Y. Qiao, and X. Tang, "Latent hierarchical model of temporal structure for complex activity classification," *IEEE Transaction on Image Processing*, vol. 23, no. 2, pp. 810–822, 2014.

[9] L. Wang, Y. Qiao, and X. Tang, "Mining motion atoms and phrases for complex action recognition," in *ICCV*, 2013.

[10] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *ICCV*, 2003, pp. 1470–1477.

[11] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: an algorithm for designing of overcomplete dictionaries for sparse representation," *TSP*, 2005.

[12] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," *NIPS*, vol. 19, p. 801, 2007.

[13] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *CVPR*, 2009.

[14] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *CVPR*, 2010.

[15] L. Liu, L. Wang, and X. Liu, "In defense of soft-assignment coding," in *CVPR*, 2011.

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

[17] X. Wang, L. Wang, and Y. Qiao, "A comparative study of encoding, pooling and normalization methods for action recognition," in *ACCV*, 2012.

[18] A. Coates and A. Y. Ng, "The importance of encoding versus training with sparse coding and vector quantization," in *ICML*, 2011.

[19] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. springer New York, 2006, vol. 1.

[20] J. van Gemert, C. Veenman, A. Smeulders, and J. Geusebroek, "Visual word ambiguity," *PAMI*, 2010.

[21] Y. Jiang, J. Yang, C. Ngo, and A. Hauptmann, "Representations of keypoint-based semantic concept detection: A comprehensive study," *TMM*, vol. 12, no. 1, pp. 42–53, 2010.

[22] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[23] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[24] T. Jaakkola, D. Haussler *et al.*, "Exploiting generative models in discriminative classifiers," *NIPS*, pp. 487–493, 1999.

[25] X. Peng, Y. Qiao, Q. Peng, and X. Qi, "Exploring motion boundary based sampling and spatial-temporal context descriptors for action recognition," in *BMVC*, 2013.

[26] L. Wang, Y. Qiao, and X. Tang, "Motionlets: Mid-level 3D parts for human motion recognition," in *CVPR*, 2013.

[27] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *IJCV*, pp. 1–20, 2013.