Multi-View Super Vector for Action Recognition

Zhuowei Cai[1], Limin Wang[1,2], Xiaojiang Peng[1], Yu Qiao[1,2*]
[1]Shenzhen Key Lab of Computer Vision and Pattern Recognition,
Shenzhen Institutes of Advanced Technology, CAS, China
[2]The Chinese University of Hong Kong, Hong Kong
{iamcaizhuowei, 07wanglimin, xiaojiangp}@gmail.com, yu.qiao@siat.ac.cn

Abstract

Images and videos can be characterized by multiple types of local descriptors such as SIFT, HOG and HOF, each of which describes certain aspects of object feature. Recognition systems benefit from fusing multiple types of these descriptors. Two widely applied fusion pipelines are descriptor concatenation and kernel average. The first pipeline is effective when different descriptors are strongly correlated, while the second one is better when descriptors are relatively independent. In practice, however, different descriptors are neither fully independent nor fully correlated, and previous fusion pipelines may not produce satisfying results. In this paper, we propose a new global representation, Multi-View Super Vector (MVSV), which is composed of relatively independent components separately encoding the shared and private information from a pair of descriptors. Kernel average is then applied to fuse these components and produce recognition results. To obtain MVSV, we develop a generative mixture model of probabilistic canonical correlation analyzers (M-PCCA). The hidden factors and gradient vectors of M-PCCA model are further utilized to construct MVSV for video representation. Experiments on video based action recognition tasks show that MVSV achieves promising results, and outperforms FV and VLAD coupled with descriptor concatenation or kernel average fusion pipeline.

1. Introduction

Action recognition has been an active research area due to its wide applications [1, 32, 33, 35]. Early research focus had been on datasets with limited size and relatively controlled settings, such as the KTH dataset [25], but later shifted to large and more realistic datasets such as the HMDB51 dataset [16] and UCF101 dataset [27]. These uncontrolled video datasets pose great challenges to the recognition task, *e.g.* large amount of intra-class variations, background clutter and occlusion, camera motions and view point changes. Recently, significant progresses have been made to improve the performance of action recognition system. These progresses should be partly ascribed to the development of more elaborately designed low-level feature descriptors [5, 19, 6], sampling strategy [32, 12] and more sophisticated developed models for generating video representations [22, 14].

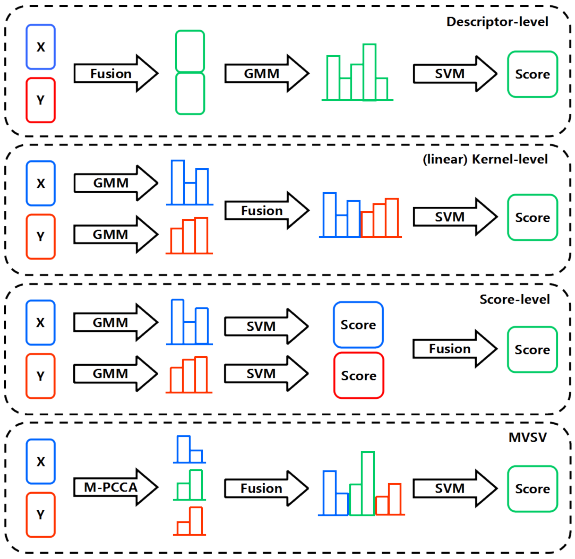


Figure 1. Different Pipelines for descriptor fusion.

To enhance recognition accuracy, several kinds of descriptors have been proposed, each of which describes certain aspects of object feature [5, 19, 32, **?**]. For example, HOG descriptor characterizes static appearance [5], while HOF and MBH descriptors capture dynamic motion [19, 6]. So far, local spatio-temporal descriptors [8] have exhibited outstanding performance [1] in action recognition task. However, successful recognition systems rarely rely on single type of descriptor. A complementary line of work focuses on local features sampling, aiming to augment rele-

*Corresponding author.

vant features and increase features coverage. ω-trajectory [12] and dense trajectory [32] are examples of practical feature sampling strategies that enhance recognition accuracy beyond the Spatio Temporal Interest Points (STIP) [18].

A standard pipeline of video encoding firstly fit the features distribution model based on the local descriptors extracted from training videos. From each new video, local features are extracted and furthered pooled into a global representation utilizing the statistics of the distribution model. Finally, this vector representation is fed into the classifier to produce recognition result. The Bag-of-Visual-Words (BoVW) representation is a classic example [26] that captures zero-th order statistics from, and implements video encoding based on the distribution of visual vocabulary. More sophisticated representations beyond BoVW have also been proposed to describe higher order statistics of features. Among them, FV [22] and its variant VLAD [14], initially designed for image classification, have been shown to achieve promising performances in several action recognition datasets [36, 21, 12].

Recent studies show that combining multiple types of local descriptors can improve recognition performance [32, 12]. Combining methods can be roughly grouped into three types, namely descriptor-level fusion, kernel-level fusion and score-level fusion, as illustrated in Figure 1. In the descriptor-level fusion, simple concatenation of weighted local descriptors is used as a new descriptor for subsequent processing [36]. The kernel-level fusion utilizes a linear combination of kernel matrices corresponding to different local descriptors to capture the structure of video data [12, 32]. Kernel average is a simple yet representative kernel-level fusion method. It is equivalent to directly concatenating the global representations corresponding to each type of descriptor, and fed the final concatenation into the linear SVM. [4] reported that kernel average is particularly effective compared to more sophisticated kernel-level fusion methods when only limited kernels are considered. The last fusion method is score-level fusion, which trains classifiers for each descriptor and fuses the confidence scores [38, 29, 37]. All these methods have been extensively evaluated in the context of complex event detection [28, 20].

Descriptor-level fusion and kernel average are widely applied in action recognition [12, 32]. When the adopted descriptors have strong mutual dependency, descriptor-level fusion is probably better, because the correlation among different descriptors are taken into account in subsequent modellings. In contrast, if different descriptors are relatively independent, kernel average will be particularly effective, because bias in one type of descriptor may be corrected by the others. However, in practice, different feature descriptors are neither fully independent nor fully correlated. As a result, different types of feature descriptors are not completely utilized, and the final recognition accuracy may deteriorate.

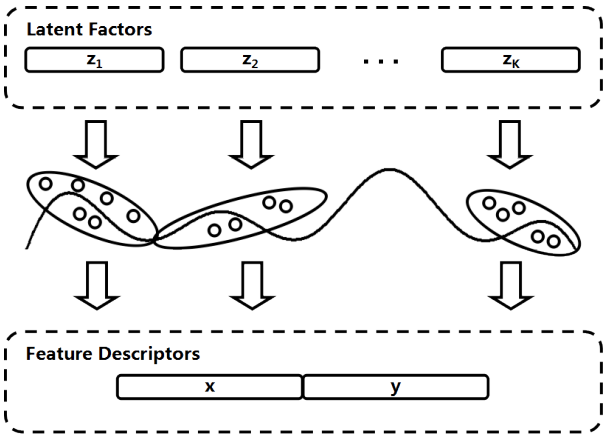


Figure 2. A graphical interpretation of M-PCCA. Each ellipse represents a PCCA submodel of M-PCCA. Observed descriptor pair is jointly generated by the mixture of K submodels.

Partly inspired by the Gaussian Mixture Model (GMM) based Fisher Vector (FV) representation [22] and the Factorized Orthogonal Latent Spaces (FOLS) approach [24] for multi-view learning, in this paper, we propose a Mixture model of Probabilistic Canonical Correlation Analyzers (M-PCCA), and utilize this model to jointly encode multiple types of descriptors for video representation. Our motivation is to factorize the joint space of descriptor pair into their shared component and mutually independent private components, so that each component has strong inner dependency while different components are as independent as possible. We then apply kernel average on these components. In this way, we make the most of different local descriptors to improve recognition accuracy. We first derive an EM algorithm to learn the parameters for M-PCCA. Each video is encoded based on this model via latent space and gradient embedding [11]. As we shall see, the resulting video representation is consisted of two components: one is the latent factors, which encodes information shared by different feature descriptors; the other is the gradient vector, which encodes information specific to each type of them. Interestingly, mathematical formulations of the two components turn out to be the counterparts of FV and VLAD representations, respectively.

The remainder of this paper is organized as follows. In Section 2, we revisit Canonical Correlation Analysis (CCA). In Section 3, we propose the mixture model of canonical correlation analyzers and its corresponding learning algorithm. Section 4 presents our video representation based on M-PCCA. An interpretation and comparison to other video representations is given at the end of Section 4. This method is experimentally examined on HMDB51 dataset and UCF101 dataset in Section 5. We conclude the paper with a discussion on the limitation and possible extension of the method.

2. Canonical Correlation Analysis revisited

In this section, we briefly review Canonical Correlation Analysis (CCA) [10] and its probabilistic extension, Probabilistic Canonical Correlation Analysis (PCCA) [3].

For two sets of data, $X = \{x_i\}_{i=1}^N$ and $Y = \{y_i\}_{i=1}^N$ with dimensions n and m respectively, CCA [10] manage to find a series of linear projections that maximize the correlation between two projected vectors aX and bY. [3] gives a probabilistic interpretation of CCA. They introduced a latent d-dimension vector z, which has a standard Gaussian distribution $p(z) = \mathcal{N}(0, I_d)$. They further assumed linear transformations between x, y and z as

$$x = W_x z + \epsilon_x, \tag{1}$$
$$y = W_y z + \epsilon_y, \tag{2}$$

where W_x and W_y are matrices with size $n \times d$ and $m \times d$ respectively. $\epsilon_x \sim \mathcal{N}(\mu_x, \Psi_x)$ and $\epsilon_y \sim \mathcal{N}(\mu_y, \Psi_y)$. x and y are assumed to be independent given the latent vector z. Intuitively, z can be considered to capture the essential information shared by both x and y while ϵ_x and ϵ_y deliver the private information specific to x and y respectively.

Given z, the conditional distributions of x and y are

$$p(x|z) = \mathcal{N}(W_x z + \mu_x, \Psi_x), \tag{3}$$
$$p(y|z) = \mathcal{N}(W_y z + \mu_y, \Psi_y). \tag{4}$$

Marginalizing over z, we have

$$p(x) = \int P(x|z)p(z)dz = \mathcal{N}(\mu_x, W_x W_x^\top + \Psi_x), \tag{5}$$
$$p(y) = \int P(y|z)p(z)dz = \mathcal{N}(\mu_y, W_y W_y^\top + \Psi_y). \tag{6}$$

The joint distribution of x and y is

$$p(x, y) = \int p(x|z)p(y|z)p(z)dz \tag{7}$$

which is Gaussian, with mean and covariance matrix

$$\mu = \begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix}, \tag{8}$$
$$\Sigma = \begin{bmatrix} W_x W_x^\top + \Psi_x, & W_x W_y^\top \\ W_y W_x^\top, & W_y W_y^\top + \Psi_y \end{bmatrix}, \tag{9}$$

3. Mixture model of Probabilistic CCA

One limitation of PCCA is that it can only deal with linear projections. This has naturally motivated researchers to develop nonlinear CCA. One example is kernel CCA [17]. An alternative paradigm to simultaneously model nonlinear structures and deal with local correlation is to introduce the mixture of local linear submodels. The effectiveness of mixture model based methods has been demonstrated in various models such as Mixture of Probabilistic Principal Component Analysis (M-PPCA) [30], mixture of experts [15], mixture of Factor analysis [9] etc. In the following section, we present our Mixture model of Probabilistic Canonical Correlation Analyzers (M-PCCA) [39] and its corresponding learning algorithm.

3.1. Model formulation

Consider a mixture model for $v = (x, y)$,

$$p(v) = \sum_k w_k p(v|k), \tag{10}$$

where the k-th submodel $p(v|k)$ is a PCCA model and $w_k = p(k)$ is its corresponding weight. Let z_k denote the latent variable in the k-th submodel. As in PCCA, conditional probability distribution on z_k can be derived as

$$p(x|k, z_k) = \mathcal{N}(W_x^k z_k + \mu_x^k, \Psi_x^k), \tag{11}$$
$$p(y|k, z_k) = \mathcal{N}(W_y^k z_k + \mu_y^k, \Psi_y^k), \tag{12}$$

where latent variable $z_k \sim \mathcal{N}(0, I_d)$ is a d-dimensional vector. Submodel $p(v|k) = \int p(v|k, z)p(z)dz$ is a Gaussian distribution $\mathcal{N}(\mu_k, \Sigma_k)$, where

$$\mu_k = \begin{pmatrix} \mu_x^k \\ \mu_y^k \end{pmatrix}, \tag{13}$$
$$\Sigma_k = \begin{bmatrix} W_x^k {W_x^k}^\top + \Psi_x^k, & W_x^k {W_y^k}^\top \\ W_y^k {W_x^k}^\top, & W_y^k {W_y^k}^\top + \Psi_y^k \end{bmatrix}. \tag{14}$$

Figure 2 gives a graphical illustration of the model. Observed data (in our case, two types of low-level feature descriptors) are sampled from a mixture model of v, where each submodel is related to a shared latent variable z_k. As we shall see, the shared information between different types of descriptor represented by z_k, and the private information encoded by ϵ_x and ϵ_y, will be utilized to construct Multi-View Super Vector for video representation.

3.2. Learning algorithm for M-PCCA

Inspired by [30], we adopt EM algorithm to learn the parameters of M-PCCA from training data $\{v_i = (x_i, y_i)\}_{i=1...,N}$. We use k-means algorithm to initialize $\mu_1, \ldots, \mu_K$ and the corresponding $\Sigma_1, \ldots, \Sigma_K$ in each local submodel. We then separately learn $\{W_x^k, W_y^k\}_{k=1,\ldots,K}$ for all K submodels via CCA using $\{\mu_k, \Sigma_k\}_{k=1,\ldots,K}$ as in [3]. In **E-step**, we estimate the responsibility $\gamma_{i,k}$ that k-th submodel contributes to generate the i-th sample for $k = 1 \ldots K$. We then compute posterior distributions of $\{z_k\}_{k=1...,K}$, which is used to estimate latent factors $\{z_{i,k}\}_{k=1,\ldots,K}$ shared by samples x_i and y_i. In **M-step**, we update $\{w_k, \mu_k, \Sigma_k, W_x^k, W_y^k\}_{k=1,\ldots,K}$ to maximize the complete data log-likelihood. Details of the EM-algorithm are provided in the Appendix. The matlab implementation of the algorithm is available at `http://zhuoweic.github.io/`.

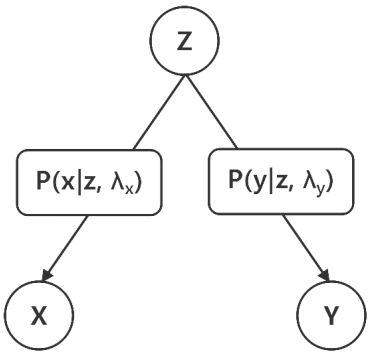


Figure 3. An interpretation of the MVSV representation. λ denotes parameters $\{\mu, \Psi, W\}$. Given z, descriptors x and y are mutually independent. z can thus be utilized as the shared information between them, with parameters λ_x and λ_y as their private information.

4. Multi-View Super Vector representation

In this section, we derive the MVSV representation from M-PCCA. Then we present an interpretation of the representation and compare our method to previous coding methods. Previous fusion methods such as descriptor concatenation and kernel average assume that different descriptors are either fully correlated or fully independent. However, this is not usually the case in real scenarios. The MVSV representation is consisted of two components: the latent factors $\mathcal{Z}$, which is the direct concatenation of estimations for z_k in each submodel, and the gradient vector $\mathcal{G}$ of the complete data log likelihood, with respect to parameters of ϵ_x and ϵ_y. This generative embedding technique originates from [11], and has been applied in Fisher Vector (FV) representation [22]. While FV embeds GMM into parameter space, we embed M-PCCA into latent space as well as parameter space. Figure 3 demonstrates our motivation to design representation with such structure. In each local PCCA, latent vector z characterizes shared information between x and y. x and y are assumed to be mutually independent given latent z. This independence is captured by their individual parameters λ_x and λ_y. Intuitively, these parameters encode private information specific to each one of them.

4.1. Constructing MVSV representation

Latent factors $\mathcal{Z}$ encoding **shared** information between x and y are firstly extracted from the M-PCCA model. For each feature sample $v_i = (x_i, y_i)$, posterior mean $\widetilde{z}_{i,k} = E(z_k|x_i, y_i)$ is used to estimate the latent vector z_k (mathematical formulation is given in the appendix). These estimations are weighted by their posterior probability, and integrated to obtain z_k using the sum-pooling scheme [36]. Specifically,

$$z_k = \left[\begin{array}{c} W_x^{k\top}, W_y^{k\top} \end{array} \right] \Sigma_k^{-1} \sum_i \gamma_{i,k}(v_i - \mu_k) \qquad (15)$$

As shown in Figure 2, our final representation of $\mathcal{Z}$ is constructed by concatenating $\{z_1, z_2, \ldots, z_K\}$. In this way, shared information between x and y is embedded into the Kd-dimensional latent space.

Gradient vectors $\mathcal{G}_x$ and $\mathcal{G}_y$ encoding **private** information specific to each type of descriptor are then derived from M-PCCA. Given latent factors $\mathcal{Z}$, conditional distributions for x and y are determined by their individual sets of parameters $\lambda_x = \{\mu_x^k, \Sigma_x^k, W_x^k\}_{k=1}^K$ and $\lambda_y = \{\mu_y^k, \Sigma_y^k, W_y^k\}_{k=1}^K$, respectively. They provide a good representation of the private information. The generative model is thus embedded into the parameter space using the gradient vector with respect to λ_x and λ_y. To keep the dimension of video representations within a reasonable size, we constrain Ψ_x^k to be diagonal, and take derivative of the complete log-likelihood of M-PCCA with respect to $\{\mu_x^k, \Psi_x^k\}$ to obtain the gradient vector $\mathcal{G}_x$. Specifically,

$$\frac{\partial E(\mathcal{L})}{\partial \mu_x^k} = -2 \sum_i \gamma_{i,k} x_{i,k} / \sigma_x^k, \qquad (16)$$

$$\frac{\partial E(\mathcal{L})}{\partial \sigma_x^k} = 2(w_k \tilde{\sigma}_x^k - \mathrm{diag}(\sum_i \gamma_{i,k} x_{i,k} x_{i,k}^\top))/\sigma_x^k \qquad (17)$$

where $w_k = \sum_i \gamma_{i,k}$, $\sigma_x^k = \sqrt{\mathrm{diag}(\Psi_x^k)}$, $\tilde{\sigma}_x^k = \sigma_x^k - \mathrm{diag}(W_x^k \Sigma_z^k W_x^{k\top})$, $x_{i,k} = x_i - \mu_x^k - W_x^k \widetilde{z}_{i,k}$. The division here is implemented as element-wise division between vectors. Gradient vector for x is thus

$$\mathcal{G}_x = \left\{ \frac{\partial E(\mathcal{L})}{\partial \mu_x^k}, \frac{\partial E(\mathcal{L})}{\partial \sigma_x^k} \right\}_{k=1,\cdots,K} \qquad (18)$$

with a dimension of Kn. Formulations for y can be derived in a similar fashion.

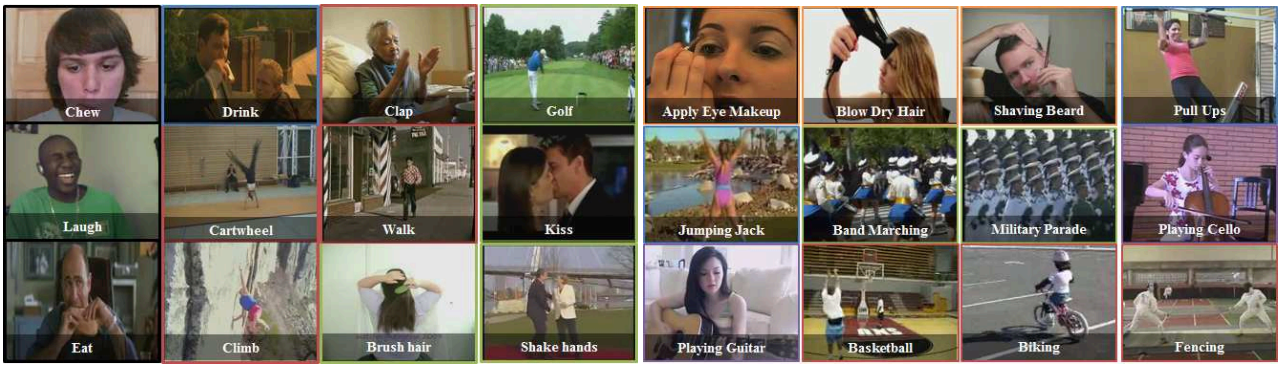
The final Multi-View Super Vector (MVSV) is constructed by concatenating the latent factors $\mathcal{Z}$ and gradient vectors $\mathcal{G}_x$ and $\mathcal{G}_y$:

$$MVSV = \{\mathcal{Z}, \mathcal{G}_x, \mathcal{G}_y\} \qquad (19)$$

This representation, with a dimension of $K(d + n + m)$, is firstly power-normalized and L2-normalized as is suggested in [23]. We then apply intra-normalization for each component as in [2].

4.2. Relation to previous methods

We concatenate the estimations of latent vector for each submodel to recover the shared information. The mathematical formulation turns out to resemble that of VLAD [14], a simplified version of FV representation. In fact, the representation (15) can be interpreted as separately applying linear transformation on each aggregated vectors of v in the k-th submodel. It has been shown that local transformation based on PCA can improve the image retrieval performance



(a) HMDB51 (b) UCF101

Figure 4. Example frames from HMDB51 dataset and UCF101 dataset.

of VLAD [7]. Our latent factors $\mathcal{Z}$ can also be extracted via a simplified version of M-PCCA. k-means is used to learn the local centroids μ_x^k, μ_y^k and corresponding local covariance matrices Ψ_x^k, Ψ_y^k. For each submodel, we then estimate W_x^k and W_y^k via CCA as in [10]. The weighting coefficient $\gamma_{t,k}$ is set to 1, if the nearest neighbour of x_t is μ_k, and 0 otherwise. Latent factors $\mathcal{Z}$ provides a natural way to compress video representation by extracting shared information from multiple types of descriptors, in that its dimension can be explicitly specified via that of the latent vector z_k. This is to be distinguished with the original VLAD representation, of which the dimension is often compressed using PCA.

As for the gradient vector $\mathcal{G}$, it can be seen as the M-PCCA counterpart of the original GMM-based Fisher Vector (FV) [22]. This gradient vector describes the direction in which parameters should be stretched to best fit the data. The difference between $\mathcal{G}$ and the original FV can be observed from equation (16) and (17), where in the case of FV, $x_{i,k} = x_i - \mu_x^k$, $\tilde{\sigma}_x^k = \sigma_x^k$ and the final super vector is normalized by the Fisher Information Matrix (FIM) [22].

5. Experiments

In this section, we first introduce the datasets used in the experiments. We then describe the implementation details including feature sampling, feature encoding and dimension specification of latent space. Then we present recognition results on HMDB51 and UCF101 databases to examine the performance of MVSV. A comparison to the state-of-the-art methods is given at the end of this section.

5.1. Datasets

The HMDB51 dataset [16] is a large collection of uncontrolled videos from various sources, including movies and YouTube videos. The dataset is consisted of 6,766 clips from 51 action categories, each category with at least 100 clips. Our experiment follows the original evaluation protocol using three sets of training/testing splits. Each split includes 70 video clips for training and 30 video clips for testing. The average accuracy for each category over the three splits is used to measure the performance.

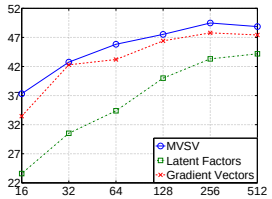
The UCF101 dataset [27] is probably the largest action recognition dataset to date. It contains 13,320 video clips collected from YouTube, with a total number of 101 action classes. Videos are grouped into 25 groups. We follow the latest released evaluation protocol with three training/testing splits [27] in the experiments and report the average accuracy. Example frames from the two datasets are shown in Figure 4.

5.2. Experimental setup

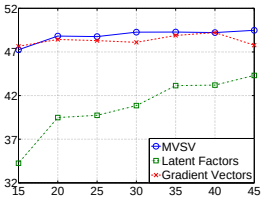
We use dense trajectory to sample local features. It has exhibited outstanding performance in action recognition task [32]. For each video, three types of features are extracted with the same setup as [32], namely Histogram of Oriented Gradient (HOG), Histogram of Optical Flow (HOF) and Motion Boundary Histogram (MBH). As in [23], we separately apply PCA on each type of descriptor to reduce their dimensions by a factor of two. We also whiten each types of descriptors as is suggested in [13]. The resulting descriptors are then L2-normalized. A total of 256,000 descriptors are randomly sampled from training data. The VLFeat Toolbox [31] is used to implement baseline methods. Specifically, we employ the built-in FV and VLAD implementations in the experiments.

5.3. Evaluation of MVSV representation

To specify the number of components and the dimension of latent space, we conduct experiments with various configurations on the second evaluation split of HMDB51 database with HOG/HOF descriptor pair. We firstly fix the dimension of latent space at 45, and vary the number of components from 16 to 512. Figure 5 (a) shows that the recognition accuracy of MVSV, latent factors and gradient vectors increase as the number of mixture components increases. This reflects the increasing effectiveness of the



(a) varied number of components



(b) varied latent dimension

Figure 5. Performance of MVSV with different configurations on the second evaluation split of HMDB51 database.

mixture model to capture the overall non-linear structure in data with increasing number of components. We also observe in Figure 5 (a) that configurations with more than 128 components result in similar performance. We fix the number of components K of GMM and M-PCCA at 256 in all the experiments for efficient implementation as well as fair comparison with other methods.

As for the dimension of latent space, we run experiments with fixed number of components at 256 and varied latent factors dimensions from 15-d to 45-d, spaced by 5-d. Figure 5 (b) demonstrates the ability of latent factors to incorporate shared information between descriptors. With increasing dimension of latent space, latent factors become more capable to capture the shared information, the gradient vectors have smaller overlap and recognition performance increases accordingly. As shown in Figure 5 (b), at 45-d, the latent factors and gradient vectors have considerable complementarity and the recognition performance reaches the peak. So we fix the latent space dimension at 45 in the subsequent experiments.

We conduct experiments to investigate the performance of MVSV representation on HMDB51. Table 1 compares recognition accuracy achieved by FV, VLAD and MVSV representation on HMDB51. The effectiveness of FV and VLAD coupled with (linear) kernel average fusion in action recognition task has been extensively investigated in [21]. As shown in Table 1, with dense trajectory, kernel average (k-level) is better than direct concatenation of descriptors (d-level) in most of the cases. HOG is in essence a descriptor of static images, while HOF and MBH are descriptors of dynamic feature. Recognition system combining HOG and HOF or MBH can probably benefit from the complementary information separately encoded in each of them.

In the first experiment on HMDB51 database, HOG is combined with MBH via several fusion methods, as shown in Table 1. MBH is treated as an entirety, which is the concatenation of independently PCAed/whitened MBHx and MBHy. MVSV achieves superior performances compared to FV and VLAD. In particular, MVSV is quite effective combining HOG and MBH. In the latter experiments, we firstly extract global representations for descriptor pairs

HMDB51	FV		VLAD		MVSV
Fusion	d-level	k-level	d-level	k-level	k-level
HOG+MBH	50.9%	50.4%	47.0%	48.5%	**52.1%**
HOG+HOF	47.0%	48.3%	44.4%	47.7%	**48.9%**
MBH(x+y)	49.2%	49.1%	45.2%	47.0%	**51.1%**
Combine	52.4%	53.2%	51.5%	52.6%	**55.9%**

Table 1. Performance of MVSV on HMDB51 database. d-level refers to direct concatenation of descriptors. k-level refers to kernel average.

UCF101	FV		VLAD		MVSV
Fusion	d-level	k-level	d-level	k-level	k-level
HOG+HOF	76.1%	77.7%	75.7%	77.5%	**78.9%**
MBH(x+y)	78.9%	78.7%	75.6%	76.3%	**80.9%**
Combine	81.1%	81.9%	80.6%	81.0%	**83.5%**

Table 2. Performance of MVSV on UCF101 database.

HOG/HOF and MBHx/MBHy, and then combine their final codes via linear kernel average. As shown in Table 1 and Table 2, the performance boost of MVSV over FV and VLAD in both HMDB51 and UCF101 database is consistent with our motivation that making each component of the representation independent helps improve recognition results.

We also conduct experiments to separately examine the recognition performance of the shared and private components of MVSV. Take the experiment on HMDB51 database combining HOG and MBH features as an example. The Latent Factors (LF) encode shared information in HOG and MBH. It has lower dimensionality (45K) compared to VLAD representation for HOG (48K) and MBH (96K). However, it achieves a comparative performance on HMDB51 database, as illustrated in Figure 6. The counterpart of FV, $\mathcal{G}_{\text{HOG}}$ (G-HOG) and $\mathcal{G}_{\text{MBH}}$ (G-MBH), achieve superior performances over FV and VLAD on HMDB51. MVSV achieves superior recognition results on HMDB51 dataset by combining the latent and gradient components.

5.4. Comparison to the state-of-the-art

Table 3 summarizes the performance of several recently published methods on HMDB51 and UCF101 datasets. Our method outperforms previous best results on HMDB51 by 1%. UCF101 is the latest released action recognition dataset. Recognition performances obtained by VLAD and FV are provided for comparisons. Our method still outperforms the best result using dense trajectory and FV by 2%.

6. Discussion and Conclusion

Video based action recognition benefits from the integration of multiple types of descriptors. In this paper, we develop mixture of probabilistic canonical correlation analyzers (M-PCCA) to jointly model the distribution of multiple types of descriptors. The mixture nature of M-PCCA allows it to capture nonlinear structure and local correla-

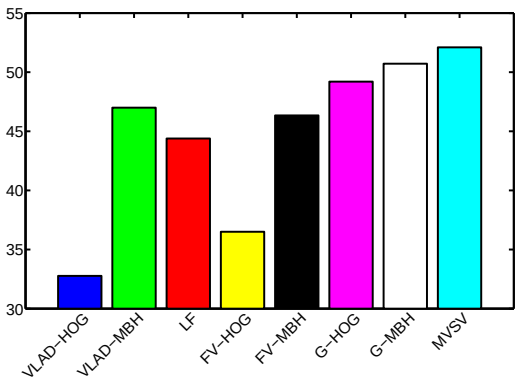


Figure 6. Performance of different components of MVSV on HMDB51. LF refers to the latent factors $\mathcal{Z}$. G-HOG and G-MBH refer to the gradient vectors $\mathcal{G}_{\text{HOG}}$ and $\mathcal{G}_{\text{MBH}}$, respectively.

HMDB51		UCF101	
STIP+BoVW [16]	23.0%	STIP+BoVW [27]	43.9%
Motionlets [34]	42.1%	DT+VLAD	79.9%
DT+BoVW [32]	46.6%	DT+FV	81.4%
FWOT [37]	48.9%		
w-traj+VLAD [12]	52.1%		
DT+FV+SPM [21]	54.8%		
MVSV	**55.9**%	MVSV	**83.5**%

Table 3. Comparison of MVSV to the state-of-the-art methods.

tion. From M-PCCA, we propose Multi-View Super Vector to integrate different types of local descriptors for action recognition. This representation consists of two relatively independent components: the latent factors encoding shared information between different types of descriptors, and the gradient vectors encoding individual information specific to each type of descriptor. Experimental results on HMDB51 and UCF101 show that the proposed MVSV representation achieves superior performance than state-of-the-art methods. However, the application of MVSV is still limited by several factors. First of all, the computation of MVSV involves matrices multiplication, which may not be affordable in recognition tasks where speed is considered critical. Second, the current framework of MVSV can only be applied on a pair of descriptors. When it comes to three or more descriptors, the fusion procedure becomes tedious. However, our framework can be extended to incorporate arbitrary number of descriptors, which will also be the focus of our future research. Note that the proposed method is not limited to action recognition. It can also be generalized to other classification tasks involving the combination of multiple feature descriptors.

7. Acknowledgement

We would like to thank the anonymous reviewers for their valuable suggestions in polishing this paper. Yu Qiao is supported by National Natural Science Foundation of China (91320101), Shenzhen Basic Research Program (JC201005270350A, JCYJ20120903092050890, JCYJ20120617114614438), 100 Talents Programme of Chinese Academy of Sciences, and Guangdong Innovative Research Team Program (No.201001D0104648280).

Appendix. EM-algorithm for M-PCCA model

$\widetilde{\gamma}_{i,k}$ and $\gamma_{i,k}$ here represent new (updated) and old (previous) variable respectively. Other variables are notated in a similar fashion. Refer to previous sections for other notations. Denote $\widetilde{x}_{i,k} = x_i - \widetilde{W}_x^k \widetilde{z}_{i,k} - \widetilde{\mu}_x^k$. Update rules for M-PCCA are as followed:

- **E-step**: update posterior distribution of the latent variables z and γ_k based on the old M-PCCA model.

$$\widetilde{\gamma}_{i,k} = \frac{w_k p(v_i|k)}{\sum_k w_k p(v_i|k)}, \tag{20}$$

$$\widetilde{z}_{i,k} = \begin{pmatrix} W_x^k \\ W_y^k \end{pmatrix}^\top \Sigma_k^{-1} \begin{pmatrix} x_i - \mu_x^k \\ y_i - \mu_y^k \end{pmatrix} \tag{21}$$

$$\widetilde{\Sigma}_z^k = I - \begin{pmatrix} W_x^k \\ W_y^k \end{pmatrix}^\top \Sigma_k^{-1} \begin{pmatrix} W_x^k, W_y^k \end{pmatrix} \tag{22}$$

$$\langle \widetilde{z}_{i,k} \widetilde{z}_{i,k}^\top \rangle = \widetilde{\Sigma}_z^k + \widetilde{z}_{i,k} \widetilde{z}_{i,k}^\top \tag{23}$$

- **M-step**: update model parameters based on the newly computed posterior distribution. For the sake of brevity, update rules for only the parameters corresponding to x are given.

$$\widetilde{w_k} = \frac{1}{N} \sum_i \widetilde{\gamma}_{i,k} \tag{24}$$

$$\widetilde{\mu}_x^k = \frac{\sum_i \widetilde{\gamma}_{i,k} x_i}{\sum_i \widetilde{\gamma}_{i,k}} \tag{25}$$

$$\widetilde{W}_x^k = \left\{ \sum_i \widetilde{\gamma}_{i,k} (x_i - \widetilde{\mu}_x^k) \widetilde{z}_{i,k}^\top \right\} \left\{ \sum_i \widetilde{\gamma}_{i,k} \langle \widetilde{z}_{i,k} \widetilde{z}_{i,k}^\top \rangle \right\}^{-1} \tag{26}$$

$$\widetilde{\Psi}_x^k = \frac{\sum_i \widetilde{\gamma}_{i,k} \widetilde{x}_{i,k} \widetilde{x}_{i,k}^\top}{\sum_i \widetilde{\gamma}_{i,k}} + \widetilde{W}_x^k \mathrm{Var}(\widetilde{z}_{i,k}) \widetilde{W}_x^{k\top} \tag{27}$$

Note that we update $\{\mu_k\}_{1,\ldots,K}$ as in [30] to simplify M-step update equations and improve speed of convergence. Parameters update rules for y can be obtained in a similar fashion.

References

[1] J. K. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *ACM Comput. Surv.*, 43(3):16, 2011.

[2] R. Arandjelovic and A. Zisserman. All about VLAD. In *CVPR*, 2013.

[3] F. R. Bach and M. I. Jordan. A probabilistic interpretation of canonical correlation analysis. *Technical Report 688, Department of Statistics, University of California, Berkeley*, 2005.

[4] S. Bucak, R. Jin, and A. Jain. Multiple kernel learning for visual object recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2013.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[6] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, 2006.

[7] J. Delhumeau, P. H. Gosselin, H. Jégou, and P. Pérez. Revisiting the vlad image representation. In *ACM Multimedia*, 2013.

[8] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005.

[9] Z. Ghahramani, G. E. Hinton, et al. The EM algorithm for mixtures of factor analyzers. 1996.

[10] D. R. Hardoon, S. Szedmák, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

[11] S. ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

[12] M. Jain, H. Jegou, and P. Bouthemy. Better exploiting motion for better action recognition. In *CVPR*, 2013.

[13] H. Jégou and O. Chum. Negative evidences and cooccurences in image retrieval: The benefit of pca and whitening. In *ECCV*, 2012.

[14] H. Jegou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, 2010.

[15] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural computation*, 6(2):181–214, 1994.

[16] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A large video database for human motion recognition. In *ICCV*, 2011.

[17] P. L. Lai and C. Fyfe. Kernel and nonlinear canonical correlation analysis. *Int. J. Neural Syst.*, 10(5):365–377, 2000.

[18] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005.

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

[20] P. Natarajan, S. Wu, S. N. P. Vitaladevuni, X. Zhuang, S. Tsakalidis, U. Park, R. Prasad, and P. Natarajan. Multimodal feature fusion for robust event detection in web videos. In *CVPR*, 2012.

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

[22] F. Perronnin and C. R. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2007.

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

[24] M. Salzmann, C. H. Ek, R. Urtasun, and T. Darrell. Factorized orthogonal latent spaces. In *AISTATS*, 2010.

[25] C. Schüldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *ICPR*, 2004.

[26] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[27] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.

[28] A. Tamrakar, S. Ali, Q. Yu, J. Liu, O. Javed, A. Divakaran, H. Cheng, and H. S. Sawhney. Evaluation of low-level features and their combinations for complex event detection in open source videos. In *CVPR*, 2012.

[29] K. D. Tang, B. Yao, F.-F. Li, and D. Koller. Combining the right features for complex event recognition. In *ICCV*, 2013.

[30] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[31] A. Vedaldi and B. Fulkerson. VLFeat: an open and portable library of computer vision algorithms. In *ACM Multimedia*, 2010.

[32] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, 2011.

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

[34] L. Wang, Y. Qiao, and X. Tang. Motionlets: Mid-level 3d parts for human motion recognition. In *CVPR*, 2013.

[35] L. Wang, Y. Qiao, and X. Tang. Latent hierarchical model of temporal structure for complex activity classification. *IEEE Transaction on Image Processing*, 23(2):810–822, 2014.

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

[37] Z. Xu, Y. Yang, I. Tsang, N. Sebe, and A. G. Hauptmann. Feature weighting via optimal thresholding for video analysis. In *ICCV*, 2013.

[38] G. Ye, D. Liu, I.-H. Jhuo, and S.-F. Chang. Robust late fusion with rank minimization. In *CVPR*, 2012.

[39] Q. Yu. Mixture model of probabilistic canonical correlation analyzers. *Technical Report, Shenzhen Institutes of Advanced Technology*, 2012.