Action Recognition via Spatio-Temporal Local Features: A Comprehensive Study

Xiantong Zhen and Ling Shao*

Department of Electronic and Electrical Engineering
The University of Sheffield

Abstract

Local methods based on spatio-temporal interest points (STIPs) have shown their effectiveness for human action recognition. The bag-of-words (BoW) model has been widely used and dominated in this field. Recently, a large number of techniques based on local features including improved variants of the BoW model, sparse coding (SC), Fisher kernels (FK), vector of locally aggregated descriptors (VLAD) as well as the naive Bayes nearest neighbor (NBNN) classifier have been proposed and developed for visual recognition. However, some of them are proposed in the image domain and have not yet been applied to the video domain and it is still unclear how effectively these techniques would perform on action recognition. In this paper, we provide a comprehensive study on these local methods for human action recognition. We implement these techniques and conduct comparison under unified experimental settings on three widely used benchmarks, *i.e.*, the KTH, UCF-YouTube and HMDB51 datasets. We discuss insightfully the findings from the experimental results and draw useful conclusions, which are expected to guide practical applications and future work for the action recognition community.

Keywords: Action Recognition, Spatio-temporal Local Features, Feature Coding, Bag-of-Words, Sparse Coding, Fisher Kernel, VLAD, NBNN, Match Kernels, Performance Evaluation

*Corresponding author (ling.shao@ieee.org)

1. Introduction

Human action recognition as an active topic in the computer vision community has been extensively researched in the last decades. Most of the existing methods, including both low-level feature extraction and high-level representations, in action recognition are extended from the text and image domains, i.e., the bag-of-word (BoW) model [1]. Local features have shown increasing effectiveness in visual recognition, and local methods based on spatio-temporal local features, *e.g.*, three-dimensional histogram of oriented gradients (HOG3D) [2] and HOGHOF [3], become popular in action recognition since the inventions of spatio-temporal interest points detectors [4, 5, 6, 7]. In contrast to holistic representations [8, 9], local methods enjoy many advantages such as 1) avoidance of some preliminary steps, *e.g.*, background subtraction and target tracking required in holistic methods, and 2) resistance to background variation and occlusions.

The most widely used local methods, *e.g.*, the bag-of-word (BoW) model [1] and sparse coding (SC) [10, 11, 12], have obtained remarkable performance in image and object classification. Recently, refinements of BoW and SC as well as alternative techniques including he soft assignment coding (kernel codebooks) [13], Triangle assignment coding [14], localized soft-assignment coding (LSC) [15] and locality linear-constrained coding (LLC) [16], have been developed to forward the state-of-the-art. However, these developments mostly remain in the image domain, which makes transferring them to the video domain an urgent and promising task.

A simple non-parametric nearest neighbor (NN) based classifier, naive bayes nearest neighbor (NBNN) [17], was proposed in recently. By computing the 'Image-to-Class' rather than 'Image-to-Image' distance, NBNN is able to avoid quantizing local features in the BoW model. In contrast to learning-based classifiers, the non-parametric NBNN classifier requires no training phase thus no risk of overfitting the parameters. Recently, enhanced versions of NBNN, including the NBNN kernels [18] and the local NBNN [19], have also been developed. The NBNN family have shown excellent effectiveness in image and object recognition.

The Fisher kernel (FK) has recently drawn increasingly attention in the image domain and produced remarkable results for image classification [20, 21, 22]. It is shown in a recent study on feature coding [23] that the improved Fisher kernel (IFK), which is also called Fisher vector (FV), outperforms all the other encoding methods on several image datasets. Another important encoding method is the vector of locally aggregated descriptors (VLAD) introduced by Jégou et al. [24, 25]. VLAD can be regarded as a simplified non-probabilistic version of Fisher

vector and has shown comparable results with IFK.

Match kernels between sets of local features have long been exploited in visual recognition [26, 27]. Without relying on any mid-level feature representations, match kernels are able to compute the similarity between sets of unordered local features and have shown the effectiveness in image and object recognition. More importantly, match kernels provides a basic formulation of measuring two sets of local features, based on which local methods are connected. The newly proposed feature coding techniques have been widely used and demonstrated their effectiveness in the image domain, however, their performance on action recognition has not been comprehensively evaluated and compared. Motivated by this, in this paper, we transfer these prevailing techniques from the image domain to the video domain and put them under a unified evaluation framework with the same experimental settings. In contrast to the previous evaluations [28, 29, 5, 30], we focus on the evaluation of state-of-the-art local methods, *e.g.*, the BoW model, sparse coding, Fisher kernels, VLAD, NBNN and match kernels, based on spatio-temporal local features for human action recognition.

Recently, methods using tracking of trajectories has been used for action recognition which can always outperform those based on STIPs while requiring higher computational complexity [31]. In addition, it is found by [32] that motion based descriptors are not scalable with respect to the number of action categories, which can be reasonably assumed to also hold for trajectory-based sampling of descriptors. As we concentrate on the comparison of representation methods rather than the overall performance, we follow a standard paradigm for action recognition using local features [28, 29], and apply the same feature detection and description steps to all the methods to be evaluated.

1.1. Contributions

We systematically evaluate the performance of representative local methods, some of which have not been used for action recognition yet. Extensive experimental results have been reported on three widely used benchmark action datasets, *i.e.*, KTH, UCF-YouTube and HMDB51. To the best of our knowledge, we, for the first time, pull local methods under a unified setting and conduct a comprehensive study both theoretically and experimentally for action recognition.

The main contributions of this paper lie in the following three aspects: **1)** we have conducted a comprehensive study on state-of-the-art local methods for human action recognition, which serves as a baseline for research in this field; **2)** we provide in-depth analysis and draw impartial conclusions from the findings in the experiments, which offers an important guide for further work on human

action recognition; **3)** we provide a timely review on the recent advancement of local methods based on spatio-temporal local features, which can be used as an up-to-date reference for the community of action recognition.

2. Review of Local Methods

During the last decade, action recognition with local spatio-temporal interest points (STIPs) have been extensively explored. To give an overview of the advancement of local features for human action recognition, we will provide a review of recently developed local methods both within and beyond the BoW model. In the following, we will give a more detailed description of these methods.

2.1. The BoW Model

The BoW model is a widely used algorithm for local representations and has proven to be successful in many action recognition tasks. However, local representations also suffer from many limitations. One of the most notorious deficiencies is that it fails to capture adequate structural and temporal information. In order to compensate for the loss of structures in local representations, a lot of methods try to improve local representations by exploring spatio-temporal structural information [33], including context information of each interest point [34, 35], relationships between/among spatio-temporal interest points [36, 37, 38, 39] and neighborhood-based features [40]. The relationship among visual words in the BoW model and their semantic meaning have also be explored to encode higher-level features [15, 41, 42, 43]. New local descriptors have also be developed [44, 45] to improve the performance of local methods.

Sun et al. [34] proposed to model the spatio-temporal context information in a hierarchical way by exploiting three levels of context, namely, point-level, intra-trajectory and inter-trajectory context. In their work, trajectories are first extracted using Scale Invariant Feature Transform (SIFT). The point-level context is the average of SIFT descriptors extracted at the salient points on the trajectory. Intra-trajectory and inter-trajectory context is modeled by the transition matrix of a Markov process and encoded as the trajectory transition and trajectory proximity descriptors.

In order to capture the most informative spatio-temporal relationship between local descriptors, Kovashka and Grauman [40] proposed to learn a hierarchy of spatio-temporal neighborhood features. The main idea is to construct a higher-level vocabulary from new features that consider the hierarchical neighboring information around each interest point.

Matikainen et al. [36] proposed to express pair-wise relationships between quantized features by combining the power of discriminative representations with key aspects of Naive Bayes. The relationship between local features is modeled as the distribution of quantized location differences between each pair of interest points. Two basic features namely STIP-HOG and quantized trajectories are considered.

Gaur et al. [33] modeled the activity in a video as a "string of feature graphs" (SFGs) by treating a video as a spatio-temporal collection of primitive features (e.g., STIP features). They divide the features into small temporal bins and represent the video as a temporally ordered collection of such feature-bins, each bin consisting of a graphical structure representing the spatial arrangement of the low-level features. A video then becomes a string of such graphs and comparing two videos is to match two strings of graphs.

Claiming that the higher-order semantic correlation between mid-level features (e.g., from the BoW representation) is useful to fill the semantic gap, Lu et al. [42] proposed novel spectral methods to learn latent semantics from abundant mid-level features by spectral embedding with nonparametric graphs and hypergraphs. A new semantics-aware representation (i.e., histogram of high-level features) is derived for each video from the original BOW representation, and actions are classified by a SVM with a histogram intersection kernel based on the new representation.

Wang et al. [38] presented a novel local representation by augmenting local features with contextual features, which capture the interactions between interest points. Different from previous work on mining contextual information is considered as spatio-temporal statistics in the 3D neighborhood of each interest point. Multi-scale channels of contextual features are computed and, for each channel, a regular grid is used to encode spatio-temporal information in the local neighborhood of an interest point. Multiple kernel learning is employed to integrate the contextual features from different channels.

Aiming to encode rich temporal ordering and spatial geometry information of local visual words, Zhang et al. [41] proposed to model the mutual relationships among visual words by a novel concept named the spatio-temporal phrase (ST phrase). A ST phrase is defined as a combination of k words in a certain spatial and temporal structure including their order and relative positions. A video is represented as a bag of ST phrases which is shown to be more informative than the BoW model.

In order to capture the geometrical distribution of interest points, Yuan et al. [39] applied the 3D R transform on the interest points based on their 3D locations.

The 3D R-transform is invariant to geometrical transformation and robust to noise. $(2D)^2$ PCA is then employed to reduce the dimensionality of the 2D feature matrix from the 3D R transform, obtaining the so-called R features. To encode the appearance features, they combined the R features with the BoW representation. Finally, they proposed a context-aware fusion method to efficiently fusion these two features. Specifically, one feature is used to compute the context of each video and the other to calculate the context-aware kernel for action recognition.

In the BoW model, mid-level features are obtained by k-means clustering which however is unable to capture the semantic relation between low-level features due to that only appearance similarity is used. Liu et al. [15] proposed to use diffusion maps to automatically learn a semantic visual vocabulary from abundant quantized mid-level features. Each mid-level feature is represented by the vector of point-wise mutual information (PMI). Diffusion maps can capture the local intrinsic geometric relations between the mid-level feature points on the manifold.

With the argument that visual words from video sequences belonging to the same class in the BoW model are correlated and jointly reflect a specific action type, Wang et al., [43], by assuming that visual words share a common structure in a low-level space, presented a framework named semi-supervised feature correlation mining (SFCM) to exploit the shared structure. A discriminative and robust classifier for action annotation is trained by taking into account the global and local structural consistency.

Shapovalova et al. [46] proposed to model a video using a global bag-of-words histogram based on local features, combined with a bag-of-words histogram focused latent regions of interest. The latent regions of interest are spatio-temporal sub-regions of a video. The model parameters are learned by a similarity constrained latent SVM, in which the constraint is to enforce that the latent regions chosen across all videos of a class are coherent.

Le et al. [44] introduced an unsupervised deep learning algorithm, named Independent Subspace Analysis (ISA), which learns spatiotemporal features of interest points from unlabeled videos. Convolution and stacking are adopted in the deep learning model to scale the algorithm to large images and learn hierarchical representations.

As indicated by Wang et al. [28] that dense sampling tends to produce better results than sparsely detected spatio-temporal interest points. Wang et al. [35] presented an approach by dense trajectories. Dense points are sampled from each frame and tracked based on displacement information from a dense optical flow field. A novel descriptor based on motion boundary histograms was introduced in their work to encode the trajectory information. The remarkable performance

of dense trajectories is largely due to the rich description of scene and contextual information of dense sampling, and the robust extraction of motion information of trajectories.

Also based on dense trajectories, Jiang et al. [47] presented a new video representation that integrates trajectory descriptors with the pair-wise trajectory locations as well as motion patterns. Global and local reference points are adopted to characterize motion information with the aim to be robust to camera movements.

2.2. Sparse Coding

Aiming to alleviate the quantization errors in the BoW model, sparse coding has also been introduced to action recognition to learn more compact and richer representations of human actions [48, 49, 12].

Rather than using the BoW model, Dean et al. [48] presented a new approach using the sparse coding algorithm to learning sparse, spatiotemporal features for activity recognition. A multi-stage approach is used to learn spatio-temporal features that can discriminate different actions.

In order to obtain a more accurate and discriminative representation, an approach by encoding local 3D spatial-temporal gradient features was proposed by Zhu et al. [49] in which the sparse coding framework is used for the final action representation. A local spatial-temporal feature is transformed to a linear combination of a few atoms in a trained dictionary. They also investigated the construction of the dictionary with a scenario of transfer learning.

Guha and Kreidieh [12] comprehensively explored sparse representations for human action recognition in video. Overcomplete dictionaries are learned from a set of local spatio-temporal descriptors in the training set. It is claimed that the obtained representation based on the dictionaries learned by sparse coding is more compact compared with the BoW model involving clustering and vector quantization. Three options of dictionaries, namely, shared, class-specific and concatenated, were investigated.

2.3. Fisher Kernels

Recently, Fisher kernels have been applied to the video domain for human action recognition based on local features. Oneata et al. [50] evaluated the use of Fisher vectors as an alternative to the BoW model to aggregate a small set of low-level descriptors, in combination with linear classifiers for both action recognition and localization. Kantorov and Laptev [51] developed highly efficient video features called the MPEG flow video descriptor using motion information in video compression and represented actions by Fisher vectors. The method improves the

speed of video feature extraction, feature encoding and action classification. Peng [52] proposed the two-layer stacked Fisher vectors (SFV) for action recognition. In the first layer, large subvolumes are densely sampled from input videos, from which local features are extracted and encoded using Fisher vectors (FVs). The second layer compresses the FVs of subvolumes obtained in the previous layer, and then encodes them again with Fisher vectors. Compared with standard FV, SFV allows refining the representation and abstracting semantic information in a hierarchical way.

2.4. Vector of Locally Aggregated Descriptors (VLAD)

Motion is regarded as the most reliable source of information for human action recognition, as it is related to the regions of interest. Jain et al. [45] introduced the Divergence-Curl-Shear (DCS) descriptor to encode scalar first-order motion features. This descriptor contains the motion divergence, curl and shear, which capture physical properties of the flow pattern. To handle the noisy motion from background and the unstable camera, an affine model is employed for motion compensation to improve the quality of descriptors. Dense trajectories are also used and the vector of locally aggregated descriptors (VLAD) is adopted for the final encoding of local features which is shown to be better than a standard BoW model. Although densely sampling shows increasing performance with the decrease of the sampling step size, it does not scale well with a large number of local patches and becomes even computationally intractable for large-scale video datasets. Vig et al. [53] proposed to select informative regions and descriptors by saliency-mapping algorithms. These regions are either used exclusively or given greater representational weights. By using the saliency-based pruning, up to 70% of descriptors can be discarded, while maintaining high performance on the Hollywood2 dataset.

2.5. Other methods

Beyond the BoW, sparse coding, FV and VLAD frameworks, many new methods have also been proposed from action representation and recognition including multiple feature fusion, matching kernels and deep learning based features.

Cai et al. [54] propose Multi-View Super Vector (MVSV) for global action representation, which is composed of relatively independent components derived from a pair of descriptors. They develop a generative mixture model of probabilistic canonical correlation analyzers (M-PCCA), and utilize the hidden factors and gradient vectors of M-PCCA to construct MVSV for video representation.

MVSV has outperforms FV and VLAD with descriptor concatenation and kernel fusion.

To encode the relationships among local feature descriptors, Wu at al. [55] construct a two-graph model based on the 3D SIFT descriptor to represent human actions by recording the spatial and temporal relationships among local features. A novel family of context-dependent graph kernels (CGKs) are further proposed to measure similarity between graphs. Finally, a generalized multiple kernel learning algorithm with a proposed $\ell_{1,2}$-norm regularization is applied to combine these CGKs optimally together and simultaneously train a set of action classifiers.

Yang and Tian [56] introduce an effective coding scheme to aggregate low-level descriptors into a super descriptor vector (SDV). In order to incorporate the spatio-temporal information, a novel approach of super location vector (SLV) was proposed to model the space-time locations of local interest points in a much more compact way compared to the spatiotemporal pyramid representations.

Sun et al. [57] propose to combine SFA with deep learning techniques to learn hierarchical representations from the video data itself. A two-layered SFA learning structure with 3D convolution and max pooling operations is used to scale up the method to large inputs and capture abstract and structural features from the video. The method shows 1% improvement in comparison to state-of-the-art methods even without supervision or dense sampling on the KTH dataset.

Recently, Lan et al. [58] propose a novel feature enhancing technique called Multi-skIp Feature Stacking (MIFS), which stacks features extracted using a family of differential filters parameterized with multiple time skips and encodes shift-invariance into the frequency space.

Yang et al. [59] propose a multi-feature max-margin hierarchical Bayesian model ($\mathrm{M^3HBM}$) for action recognition. $\mathrm{M^3HBM}$ jointly learns a high-level representation by combining a hierarchical generative model (HGM) and discriminative max-margin classifiers in a unified Bayesian framework.

3. Methods

In this section, we describe the widely-used methods based on local features for visual recognition which will be evaluated in this work.

3.1. The BoW Model

Local features in the training set are first clustered to create a codebook [60]. Video sequences are represented by coding local features with the visual words in the codebook. The coding methods to be used in the BoW model include the

hard assignment, the soft assignment [13], the triangle assignment [14] and the localized soft assignment [61].

Before describing the details of all the coding methods, we first define the notations used in both the BoW model and sparse coding (SC). Let $\mathbf{b}_i$ denote a visual word or a basis vector, and $\mathbf{B}_{D\times M}$ denote a codebook or a set of basis vectors, where D is the dimensionality of the local feature vectors and M is the number of codewords or bases. $\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_N$ are local features from a video sequence, $\mathbf{u}_i \in R^M$ is the coding coefficient vector of $\mathbf{x}_i$ based on the codebook or basis vectors. u_{ij} is the coefficient associated with the word $\mathbf{b}_j$.

3.1.1. Hard Assignment

In the hard assignment coding, the coefficient of each local feature is determined by assigning this feature $\mathbf{x}_i$ to its nearest codeword in the codebook using a certain distance metric. If the Euclidean distance is used, then

$$u_{i,j} = \begin{cases} 1 & \text{if } j = \arg\min_{j=1,\cdots,M} ||\mathbf{x}_i - \mathbf{b}_j||_2^2 \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

3.1.2. Soft Assignment

In the soft assignment coding, The coefficient $u_{i,j}$ is the degree of membership of a local feature $\mathbf{x}_i$ to the $j$thcodeword.

$$u_{ij} = \frac{\exp(-\beta||\mathbf{x}_i - \mathbf{b}_j||_2^2)}{\sum_{k=1}^{M} \exp(-\beta||\mathbf{x}_i - \mathbf{b}_k||_2^2)} \tag{2}$$

where β is the smoothing factor controlling the softness of the assignment.

3.1.3. Triangle Assignment

The triangle assignment coding was proposed in [14]. The coding is defined by the following activation function:

$$u_{ij} = \max\{0, \mu(\mathbf{z}) - z_j\} \tag{3}$$

where $z_j = ||\mathbf{x}_i - \mathbf{b}_j||_2$ and $\mu(\mathbf{z})$ is the mean of elements of $\mathbf{z}$. This activation function forces the output to be 0 for any feature $\mathbf{x}_i$ whose distance to the codeword b_j is larger than the average of all distances. As a result, roughly half of the weights will be set to 0.

3.1.4. Localized Soft assignment Coding (LSC)

By combining the ideas of localization and the soft assignment coding, Liu *et al.*[61] proposed the localized soft-assignment coding (LSC). The activation function takes the form in Equation 2, but with the locality constraint as follows:

$$d(\mathbf{x}_i, \mathbf{b}_j) = \begin{cases} d(\mathbf{x}_i, \mathbf{b}_j), & \text{if } \mathbf{b}_j \in N_k(\mathbf{x}_i) \\ \infty & \text{otherwise.} \end{cases}, \tag{4}$$

where $d(\mathbf{x}_i, \mathbf{b}_j) = ||\mathbf{x}_i - \mathbf{b}_j||_2^2$, and N_k denotes the k-nearest neighbors of $\mathbf{x}_i$ defined by the distance $d(\mathbf{x}_i, \mathbf{b}_j)$.

3.2. Sparse Coding

In sparse coding (SC), a local feature is represented by a linear combination of a sparse set of basis vectors. The coding coefficient is obtained by solving an l_1-norm regularized approximation problem [62]:

$$\mathbf{u}_i = \arg\min_{u \in R^n} ||\mathbf{x}_i - \mathbf{B}\mathbf{u}||_2^2 + \lambda ||\mathbf{u}||_1, \tag{5}$$

where λ controls the sparsity of the coefficient.

3.2.1. Locality-constrained Linear Coding (LLC)

Instead of enforcing sparsity in SC, LLC [16] confines a local feature $\mathbf{x}_i$ to be coded by its local neighbors in the codebook. The locality constraint ensures that similar patches would have similar codes. The coding coefficient is obtained by solving the following optimization problem:

$$\begin{aligned} \mathbf{u}_i = \arg\min_{\mathbf{u} \in R^M} & ||\mathbf{x}_i - \mathbf{B}\mathbf{u}||_2^2 + \lambda ||\mathbf{d}_i \odot \mathbf{u}||_2^2, \\ & s.t. \quad \mathbf{1}^T \mathbf{u}_i = 1 \end{aligned} \tag{6}$$

where $\odot$ denotes the element-wise multiplication, and $\mathbf{d}_i \in R^M$ is the locality adaptor that gives different freedom for each basis vector proportional to its similarity to the input descriptor $\mathbf{x}_i$. Specifically,

$$\mathbf{d}_i = \exp(-\frac{dist(\mathbf{x}_i, \mathbf{B})}{\sigma}) \tag{7}$$

where $dist(\mathbf{x}_i, \mathbf{B}) = [dist(\mathbf{x}_i, \mathbf{b}_1), ..., dist(\mathbf{x}_i, \mathbf{b}_M)]^\top$, and $dist(\mathbf{x}_i, \mathbf{b}_j)$ is the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{b}_j$. σ is used for adjusting the weight decay

speed for the locality adaptor. As an approximation of LLC, one can simply use the k nearest neighbors of $\mathbf{x}_i$ as the local bases, and solve a much smaller linear system.

3.3. Improved Fisher Kernel (IFK)

Fisher kernels [63] was introduced to combine the advantages of both generative and discriminative models. Perronnin et al. [20] applied Fisher kernels to learn a visual vocabulary for image representation. An image is described with a gradient vector derived from its probability density function.

Specifically, $X = \{\mathbf{x}_n, n = 1 \dots N\}$ denotes the set of low-level feature vectors extracted from an image and λ is the set of parameters of the Gaussian mixture model (GMM). $\lambda = \{\omega_i, \mu_i, \Sigma_i, i = 1 \dots M\}$ where ω_i, μ_i and Σ_i denote respectively the weight, mean vector and covariance matrix of Gaussian i and M denotes the number of Gaussians.

With the assumption that local features are independent, an image can be represented by the likelihood of all the local features as:

$$L(X|\lambda) = \sum_{n=1}^{N} \log p(\mathbf{x}_n|\lambda) \tag{8}$$

where $p(x_n|\lambda)$ is the probability density function that can be modeled by the GMM model. The image can be described by the gradient vector:

$$G = \frac{1}{N} \nabla_\lambda L(X|\lambda) \tag{9}$$

The gradient of the likelihood describes the contribution of the parameters to the generation process [21]. A kernel between two gradient vectors of images X and Y is

$$K(X, Y) = (G_\lambda^X)^\top \mathcal{F}_\lambda^{-1} G_\lambda^Y \tag{10}$$

where $\mathcal{F}_\lambda^{-1}$ is the Fisher information matrix

$$\mathcal{F}_\lambda = E[\nabla_\lambda L(X|\lambda) \nabla_\lambda L(Y|\lambda)] \tag{11}$$

The Fisher information matrix $\mathcal{F}_\lambda$ is symmetric and positive definite, and has a Cholesky decomposition $\mathcal{F}_\lambda = F_\lambda^\top F_\lambda$. X is then can be represented by a normalized gradient vector:

$$\mathcal{G}_\lambda^X = F_\lambda G_\lambda^X \tag{12}$$

which is referred as the Fisher vector of X.

According to [20], the Fisher matrix has an approximated closed-form solution with which the Fisher vector can be represented as: $v(i) = [\mathcal{G}_{\mu,i}; \mathcal{G}_{\Sigma,i}]$, where i indexes the i-th Gaussian of the Fisher vector and

$$\mathcal{G}_{\mu,i} = \frac{1}{N\sqrt{\omega_i}} \sum_{n=1}^{N} \gamma_n(i)(\frac{\mathbf{x}_n - \mu_i}{\sigma_i}) \tag{13}$$

$$\mathcal{G}_{\sigma,i} = \frac{1}{N\sqrt{2\omega_i}} \sum_{n=1}^{N} \gamma_n(i)[\frac{(\mathbf{x}_n - \mu_i)^2}{\sigma^2} - 1] \tag{14}$$

where $\gamma_i(n) = \frac{\omega_i p_i(x_n|\lambda)}{\sum_{j=1}^{M} \omega_i p_j(x_n|\lambda)}$. We use the improved Fisher kernel (IFK) proposed in [21] which has shown to significantly improve the original Fisher kernel.

3.4. Vector of Locally Aggregated Descriptors (VLAD)

The vector of locally aggregated descriptors (VLAD) was proposed by Jégou et al. [24, 25] which is a simplified non-probabilistic version of the Fisher vector. To be consistent with the BoW method, $\mathbf{B} = \{\mathbf{b}_i, i = 1, \ldots, M\}$ is the codebook. Each local descriptor $\mathbf{x}_n$ is associated with its nearest visual word $NN(\mathbf{x}_n)$ in the codebook. For each codeword $\mathbf{b}_i$, the differences $\mathbf{x}_n - \mathbf{b}_i$ of the vectors $\mathbf{x}_i$ assigned to $\mathbf{b}_i$ are accumulated:

$$\mathbf{v}_i = \sum_{\mathbf{x}_n : NN(\mathbf{x}_n) = i} (\mathbf{x}_n - \mathbf{b}_i) \tag{15}$$

The concatenation $\mathbf{v} = [\mathbf{v}_1, \ldots, \mathbf{v}_i, \ldots, \mathbf{v}_M,]$ is the final compact representation of an image/video.

3.5. Match Kernels

Match kernels between sets of local features have long been exploited [27, 26]. The kernel function is computed to measure the similarity between two images/video sequences represented by sets of local feature vectors.

Given two feature sets, $\mathcal{F}_a = \{F_1^{(a)}, \ldots, F_{|\mathcal{F}_a|}^{(a)}\}$ and $\mathcal{F}_b = \{F_1^{(b)}, \ldots, F_{|\mathcal{F}_b|}^{(b)}\}$, the summation kernel is defined as:

$$K_S(\mathcal{F}_a, \mathcal{F}_b) = \frac{1}{|\mathcal{F}_a|} \frac{1}{|\mathcal{F}_b|} \sum_{i=1}^{|\mathcal{F}_a|} \sum_{j=1}^{|\mathcal{F}_b|} K_F(F_i^{(a)}, F_j^{(b)}) \tag{16}$$

In [27], a kernel function (the max-sum kernel) for matching local features was proposed:

$$K_M(\mathcal{F}_a, \mathcal{F}_b) = \frac{1}{2}\sum_{i=1}^{|\mathcal{F}_a|} \max_{j=1,\ldots,|\mathcal{F}_b|} K_F(F_i^{(a)}, F_j^{(b)}) + \frac{1}{2}\sum_{j=1}^{|\mathcal{F}_b|} \max_{i=1,\ldots,|\mathcal{F}_a|} K_F(F_j^{(b)}, F_i^{(a)}) \quad (17)$$

This match kernel has been used in object recognition [27] and action classification [64]. Lyu *et al.* [26] has proven it to be a non-mercer kernel, and proposed a normalized sum-match kernel which satisfies the mercer condition and is defined as follows:

$$K_{\mathcal{F}}(\mathcal{F}_a, \mathcal{F}_b) = \frac{1}{|\mathcal{F}_a|}\frac{1}{|\mathcal{F}_b|}\sum_{i=1}^{|\mathcal{F}_a|}\sum_{j=1}^{|\mathcal{F}_b|}[K_F(F_i^{(a)}, F_j^{(b)})]^p, \quad (18)$$

where $p \geq 1$ is the kernel parameter.

3.6. Naive Bayes Nearest Neighbor (NBNN)

Naive Bayes Nearest Neighbor (NBNN) is an approximation of the optimal MAP Naive-Bayes classifier. Given an image Q represented as a set of local features, $\mathbf{x}_1, \ldots, \mathbf{x}_N$, when the class prior $p(C)$ is uniform, MAP becomes the maximum likelihood (ML) classifier:

$$\hat{C} = arg\max_C p(C|Q) = arg\max_C p(Q|C). \quad (19)$$

With the Naive-Bayes assumption that $\mathbf{x}_1, \ldots, \mathbf{x}_N$ are i.i.d. given its class C, we have

$$p(Q|C) = p(\mathbf{x}_1, \ldots, \mathbf{x}_N|C) = \prod_{i=i}^{N} p(\mathbf{x}_i|C) \quad (20)$$

$p(\mathbf{x}_i|C)$ is further approximated using the Parzen density estimation and when the Parzen kernel keeps only the nearest neighbor and the same kernel bandwidth for

all the classes, the resulting classifier takes the following simple form:

$$\bar{c} = arg\min_{c} = \sum_{\mathbf{x}\in\mathbf{X}} ||\mathbf{x} - NN^c(\mathbf{x})||^2, \tag{21}$$

where NN^c is the nearest neighbor of $\mathbf{x}$ in class c.

3.6.1. NBNN Kernels

The NBNN kernel is based on the normalized sum match kernel [26], to calculate the similarity between two sets of features $X = \{\mathbf{x}\}$ and $Y = \{\mathbf{y}\}$:

$$\begin{aligned} K(X,Y) &= \sum_{c\in C} K^c(X,Y) \\ &= \frac{1}{|X||Y|} \sum_{c\in C} \sum_{\mathbf{x}\in X} \sum_{\mathbf{y}\in Y} k^c(\mathbf{x},\mathbf{y}), \end{aligned} \tag{22}$$

where $C = \{c\}$ and $k^c(\mathbf{x},\mathbf{y})$ is the local kernel between local features $\mathbf{x}$ and $\mathbf{y}$. In the NBNN kernel, $k^c(\mathbf{x},\mathbf{y})$ is defined as:

$$\begin{aligned} k^c(\mathbf{x},\mathbf{y}) &= \phi^c(\mathbf{x})^{\mathrm{T}} \phi^c(\mathbf{y}) \\ &= f^c(d^1_{\mathbf{x}},\ldots,d^{|C|}_{\mathbf{x}})^{\mathrm{T}} f^c(d^1_{\mathbf{y}},\ldots,d^{|C|}_{\mathbf{y}}) \end{aligned} \tag{23}$$

Two distance functions have been considered in the original work [18], namely, $f^c_1(d^1_{\mathbf{x}},\ldots,d^{|C|}_{\mathbf{x}}) = d^c_{\mathbf{x}}$ and $f^c_2(d^1_{\mathbf{x}},\ldots,d^{|C|}_{\mathbf{x}}) = d^c_{\mathbf{x}} - d^{\hat{c}}_{\mathbf{x}}$, where $d^c_{\mathbf{x}}$ is the distance to its nearest neighbor in class c and $d^{\hat{c}}_{\mathbf{x}}$ denotes the closest distance to all classes except for c.

3.6.2. Local NBNN

McCann and Lowe [19] developed an improved version of NBNN, named local naive bayes nearest neighbor (LNBNN), which increases the classification accuracy and scales better with a large number of classes. The motivation of local NBNN is from the observation that only the classes represented in the local neighborhood of a descriptor contribute significantly and reliably to their posterior probability estimation. Instead of finding the nearest neighbor in each of the classes, local NBNN finds in the local neighborhood k nearest neighbors which may only come from some of the classes. The "localized" idea is shared with LSC in the BoW model and LLC in SC.

4. Experiments and Results

To comprehensively investigate and evaluate local methods for human action recognition, we have conducted extensive experiments on three increasingly challenging benchmarks including the KTH, UCF-Youtube and HMDB51 datasets.

4.1. Datasets

The KTH dataset [65] is a commonly used benchmark action dataset with 2391 video clips. Six human action classes, including walking, jogging, running, boxing, hand waving and handicapping, are performed by 25 subjects in four different scenarios: outdoors (s1), outdoors with scale variation (s2), outdoors with different clothes (s3) and indoors with lighting variation (s4). We follow the standard experimental setup [28], *i.e.*, test set (9 subjects: 2, 3, 5, 6, 7, 8, 9, 10, and 22) and training set (the remaining 16 subjects).

The UCF YouTube dataset [15] contains 11 action categories of 11 action categories: basketball shooting, biking/cycling, diving, golf swinging, horse back riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog. This dataset is challenging due to large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background and illumination condition. We follow the experimental settings in [15].

The HMDB51 dataset [66] contains 51 distinct categories with at least 101 clips in each for a total of 6766 video clips extracted from a wide range of sources. The action categories can be grouped in five types: 1) General facial actions: smile, laugh, chew, talk; 2) Facial actions with object manipulation: smoke, eat, drink; 3) General body movements: cartwheel, clap hands, climb, climb stairs, dive, fall on the floor, backhand flip, handstand, jump, pull up, push up, run, sit down, sit up, somersault, stand up, turn, walk, wave; 4) Body movements with object interaction: brush hair, catch, draw sword, dribble, golf, hit something, kick ball, pick, pour, push something, ride bike, ride horse, shoot ball, shoot bow, shoot gun, swing baseball bat, sword exercise, throw; 5) Body movements for human interaction: fencing, hug, kick someone, kiss, punch, shake hands, sword fight. All the results are reported by averaging the three training/test splits [66].

4.2. Experimental settings

In this section, we give the implemental details and parameter settings of each method evaluated in our experiments.

4.2.1. Spatio-temporal local features

We employ the periodic detector proposed by Dollár at al. [67] to detect the spatio-temporal interest points from the raw video sequences and follow the parameter settings in the evaluation work of [28]. As in [14], the three-dimensional histogram of oriented gradients (HOG3D) [2] is used to describe each STIP due to its computational efficiency. The chosen detector and descriptor have shown outstanding performance in [28, 29]. For BoW and SC, we randomly select 100000 local features from the training set to learn codebooks and dictionaries.

The spatio-temporal pyramid matching (STPM) [68] can be easily embedded in the methods to encode the structural information and presumably could improve the performance. As our focus is on the comparison between different methods rather than the overall performance, and we argue that STPM would equally contribute to each method, STPM is not used in our evaluation framework.

4.2.2. Feature Pooling

In BoW and SC, a final representation $\mathbf{v}$ of an action is obtained by pooling over the coefficients [11]. With average pooling, the j-th component of $\mathbf{v}$ is obtained by $v_j = \sum_{i=1}^{N} u_{ij}/N$. With max pooling, v_j is obtained by $v_j = \max_i u_{ij}$, where $i = 1, 2, \cdots, N$.

4.2.3. The BoW Model

In the BoW model, the codebooks are created by the k-means clustering algorithm provided in VLFeat toolbox [60] with a single run and is fixed for encoding methods under the BoW framework. In LSC, we follow the parameter settings in the original work [61] with β set as 10. For hard assignment coding, we has also implemented square rooting with l2 normalization [22].

4.2.4. Sparse Coding

For sparse coding, we use the open-source optimization toolbox SPAMS (SPArse Modeling Software) [1]. The dictionary is learned by the algorithm in [62], and the sparse codes are learned using orthogonal matching pursuit (OMP) [62]. The parameter λ is set 0.15. The number of non-zero coefficients is 10 in the OMP algorithm. For LLC, we use the released code with the same parameter settings.

[1] http://spams-devel.gforge.inria.fr/

4.2.5. NBNN

As NBNN is non-parametric, no parameter is required to tune. While for the local NBNN classifier, the single parameter is the number of nearest neighbors k. We have investigated the effect of k in our experiments. With regard to the NBNN kernel, we have experimented the distance function $f_2^c(d_{\mathbf{x}}^1, \ldots, d_{\mathbf{x}}^{|C|})$ in our implementation.

4.2.6. Match Kernels

For the match kernels, we use the linear kernel as the local kernel and the single parameter p is set as 9 according to the original work [26]. We also use the normalized kernel in building the SVM classifier: $K(x, y) \leftarrow \frac{K(x,y)}{\sqrt{K(x,x)}\sqrt{K(y,y)}}$.

4.2.7. IFK

We use the implementation of the Fisher vector provided by the VLFeat toolbox [60]. The effect of different numbers of Gaussians has also been investigated. We follow [20, 21] by using the GMM to model the probability density function $p(\mathbf{x}|\lambda)$ in Eq. (8).

4.2.8. VLAD

We also use the implementation of VLAD by the VLFeat toolbox [60].Similar to IFK, L2-normalization with square rooting is used to improve the performance. We use the results (the means of GMM) of GMM in IFK as the codebook for VLAD.

4.2.9. Action Classification

With the final action representation, we use a support vector machine (SVM) [69] classifier for BoW, SC, the improved Fisher kernel, VLAD and the match kernels. The performance of different kernels has also been evaluated. Note that the χ^2 and intersection kernels are only applicable to histogram representations. For BoW and SC, we have also experimented with different kernels of SVMs. The recognition performance is measured by classification accuracy.

4.3. Results

All the final results on the three datasets are summarized in Table 1. The size of the codebook in BoW and the number of bases in SC which are hard to pre-determine while always affect the performance have been investigated and illustrated.

Methods	KTH	YouTube	HMDB51
BoW-Hard	87.9%	58.1%	20.0%
BoW-Hard (Sqrt-L2-Normalization)	91.8%	59.5%	23.5%
BoW-Soft-Average	85.4%	53.5%	19.6%
BoW-Soft-Max	89.2%	61.2%	24.0%
BoW-Triangle-Average	84.1%	52.5%	20.7%
BoW-Triangle-Max	89.8%	61.0%	25.1%
BoW-LSC	92.5%	59.4%	24.6%
SC-Average	91.0%	56.0%	23.3%
SC-Max	91.5%	59.4%	27.9%
SC-LLC	91.3%	56.2%	24.1%
NBNN	93.9%	57.8%	19.8%
NBNN Kernel	89.2%	62.4%	23.7%
Local NBNN	**94.1%**	60.1%	21.2%
VLAD	92.0%	62.6%	26.4%
Improved Fisher Kernel	93.2%	**63.0%**	**30.5%**
Match Kernel	86.9%	54.5%	13.7%
State-of-the-Art*	90.0% [28] 97.0% [55]	61.7% [70] 93.38%[52]	30.1% [70] 63.9% [58]

*__The upper row__: results based on the same experimental settings, *i.e.*, spatio-temporal interest point detector and HOG3D descriptor to the evaluation setting in this work. **The bottom row**: results of the up-to-date methods based on more sophisticated features and learning techniques.

Table 1: The performance comparison of all methods on three datasets, *i.e.*, KTH, UCF-YouTube and HMDB51. Note that the results of the match kernel are obtained by $K_{\mathcal{F}}$ (recognition rate).

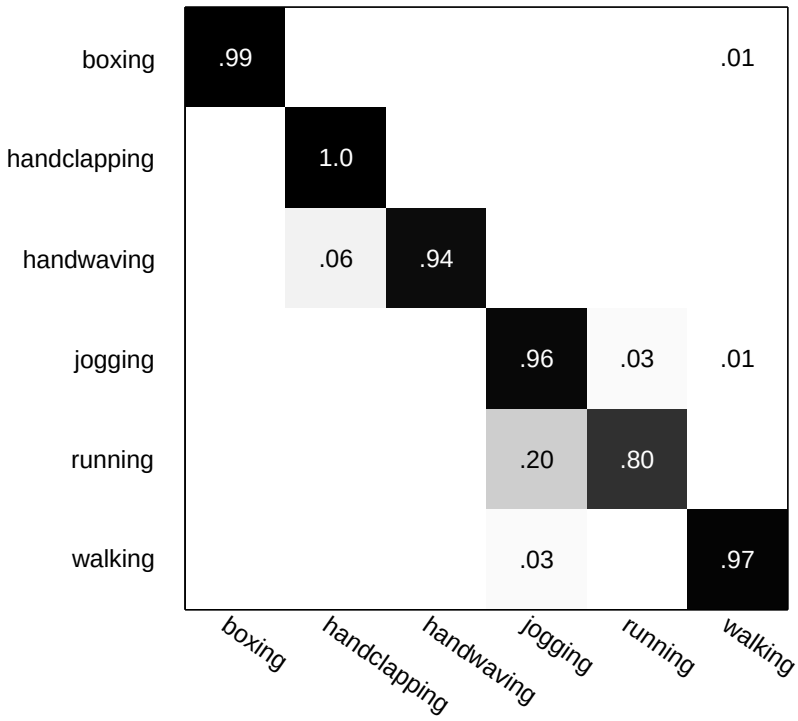

Figure 1: The confusion matrix of the results with local NBNN on the KTH dataset (recognition rate).

4.3.1. On the KTH dataset

The best result is 94.1% obtained by the local NBNN classifier, which is comparative to the state-of-of-art results from more complicated methods. The confusion matrix of the best result is plotted in Fig. 1. The NBNN classifier achieves the second best result - 93.9% - which is slightly lower than the local NBNN classifier. In addition, the NBNN kernel gives a result of 89.2%, which is still better than the baseline hard assignment coding in BoW.

In the BoW model, LSC achieves an accuracy of 92.5% which is impressive considering its simplicity. The triangle assignment coding with max pooling is better than both the hard and soft assignment coding techniques, which is consistent with the report in [14]. The effect of kernels on different methods has also

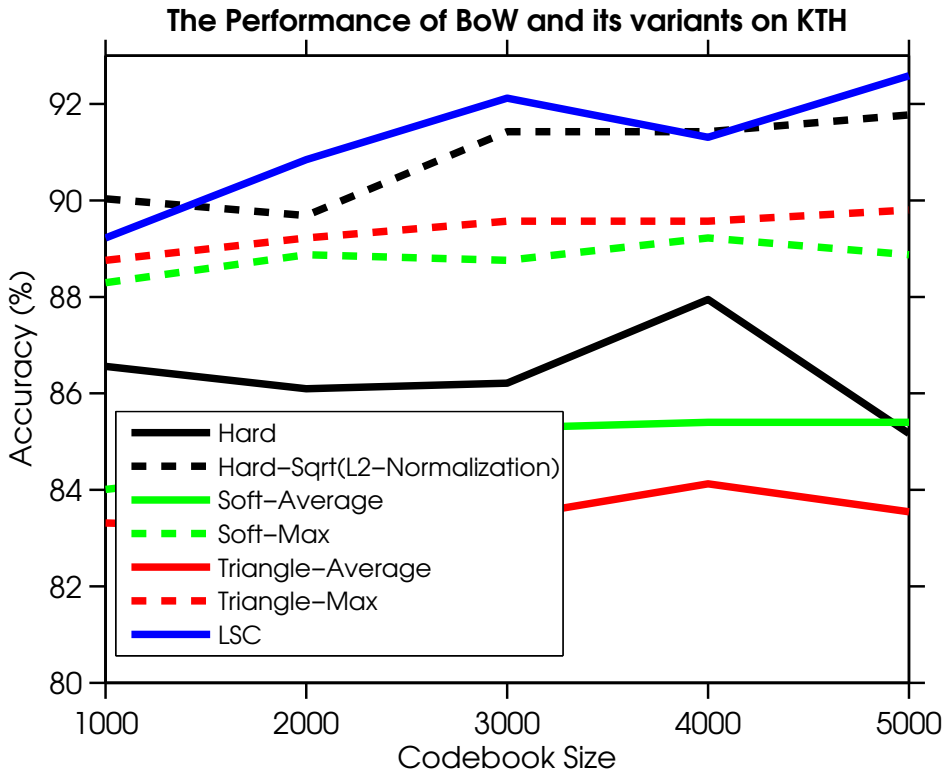

Figure 2: The performance of the BoW model and its variants on the KTH dataset (recognition rate).

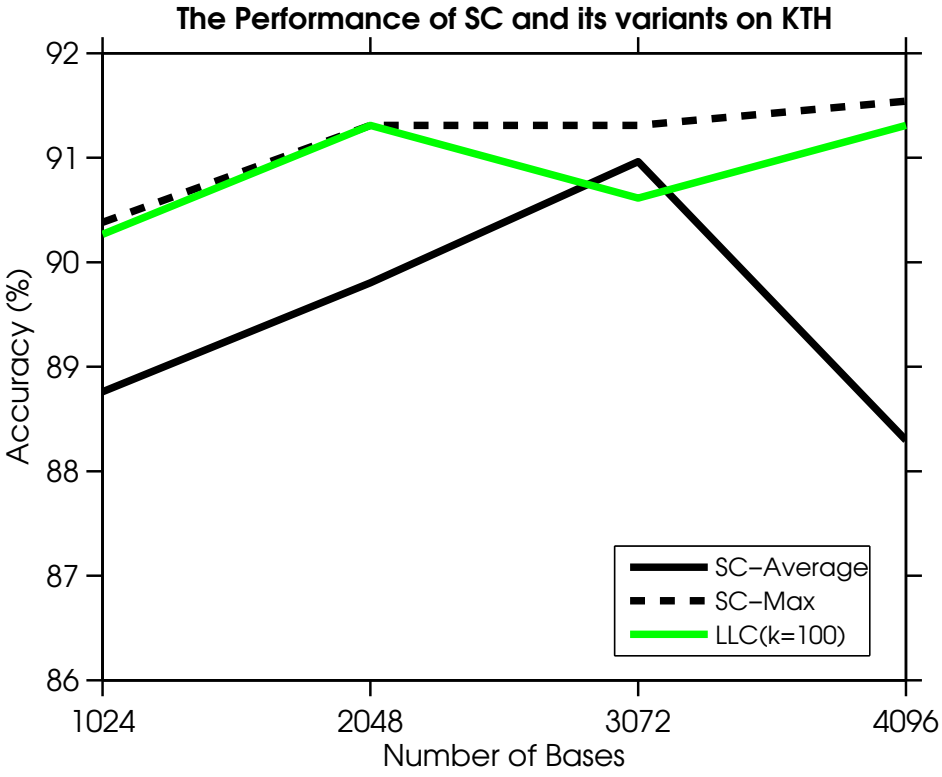

Figure 3: The performance of SC and its variants on the KTH dataset (recognition rate).

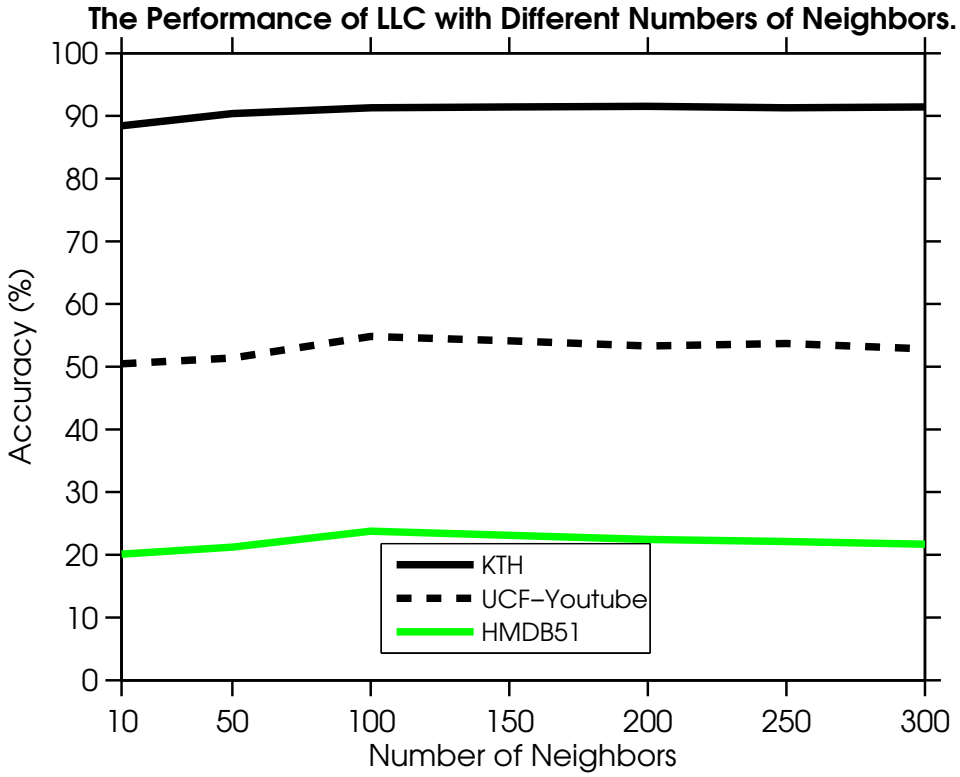

Figure 4: The performance of LLC with different numbers of neighbors (recognition rate).

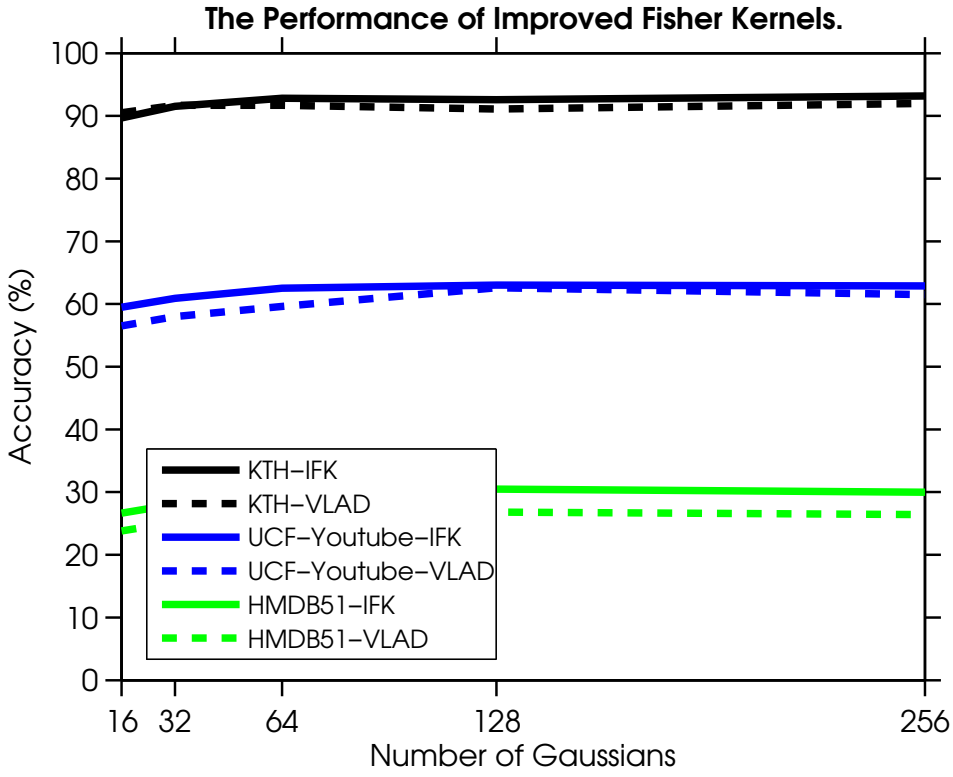

Figure 5: The performance of IFK with different numbers of Gaussians (recognition rate).

Kernels	Linear	RBF	χ^2	Intersection
BoW-Hard (Sqrt-L2-Normalization)	91.8%	91.0%	91.7%	90.8%
BoW-Soft-Max	88.9%	84.4%	-	-
BoW-Triangle-Max	89.8%	89.8%	-	-
BoW-LSC	92.5%	90.7%	-	-
SC-Max	91.5%	90.3%	-	-
LLC	91.3%	89.1%	-	-

Table 2: The performance of SVM with different kernels on KTH (recognition rate).

been investigated and the results on KTH are reported in Table 2. Note that our implementation of the baseline hard assignment coding with the χ^2 kernel is better than that in [28] (91.7% vs. 90.0%). Among all the kernels, the linear kernel produces the best performance on this dataset and the intersection kernel is also comparable with other kernels. The effect of codebook sizes on the BoW model is illustrated in Fig. 2. Most of the methods peak around 4000 codewords except for LSC which keeps increasing up to 5000 codewords.

The SC-based methods yield relatively better results compared with the BoW model. The ordinary SC with max pooling achieves even better results than LLC. Both SC and LLC reach the best results around 3072 bases as shown in Fig. 3. The number k of nearest neighbors in LLC is a key parameter in LLC and the performance with different values of k is shown in Fig. 4. The performance becomes relatively stable with $k > 100$. The linear kernel outperforms radial basis function (RBF) kernel within SC as shown in Fig. 2. The results in [48] and [49] which use sparse coding on this dataset are 85,73% and 94.92%. A different experimental setting, i.e., using 599 video clips in total, is employed in [49] for validation.

The improved Fisher kernel (IFK) has achieved a high accuracy on this dataset which is better than both BoW and SC based methods. IFK even outperforms the NBNN kernel and is comparable with NBNN and local NBNN. The performance of VLAD is also impressive with an accuracy of 92.0%. The performance of IFK and VLAD with different numbers of Gaussians is shown in Fig. 5. The match kernel performs poorly with an accuracy of 86.9%.

4.3.2. On the UCF-YouTube dataset

The results on the UCF-YouTube dataset are slightly different from those on the KTH dataset. Among the NBNN methods, the NBNN kernel produces the best result of 62.4% which is slightly better than local NBNN. The corresponding

Kernels	Linear	RBF	χ^2	Intersection
BoW-Hard (Sqrt-L2-Normalization)	59.5%	45.5%	53.2%	59.8%
BoW-Soft-Max	59.4%	47.1%	-	-
BoW-Triangle-Max	61.0%	49.7%	-	-
BoW-LSC	59.4%	48.3%	-	-
SC-Max	59.4%	49.5%	-	-
LLC	53.9%	47.8%	-	-

Table 3: The performance of SVM with different kernels on UCF-Youtube (recognition rate).

confusion matrix is plotted in Fig. 8.

In the BoW model, the soft assignment coding with max pooling performs best which is better than the triangle assignment coding and LSC. The result -61.2%- is comparable with theresult -62.4%- by the NBNN kernel. As shown in Fig. 6, the best results happen around 5000 codewords for almost all the methods within the BoW framework. SC with max pooling outperforms LLC obtaining an accuracy of 59.4% which is also comparable with the best result. The effect of different numbers of bases in SC is illustrated in Fig. 7, and most of the best results for SC and LLC occur with 4096 bases. The effect of kernels on the performance of BoW and SC is reported in Table 3. On this dataset, the intersection kernel outperforms the linear kernel within BoW and the linear kernel is significantly better than the RBF kernel within SC. In addition, the performance variation of LLC with the number of neighbors is illustrated in Fig. 4. The performance of SVM with different kernels on this dataset is reported in Table 6.

The improved Fisher kernel has achieved the best performance -63.0%- on this dataset which is slightly better than that -62.6%- of VLAD . The effect of different numbers of Gaussians on this dataset is also shown in Fig. 5. The performance of the match kernels is inferior in this dataset, producing a low recognition rate of 54.5%.

4.3.3. On the HMDB51 dataset

The best result -30.5%- is obtained by the improved Fisher kernel which is better than the rest of the evaluated methods with a large margin. The confusion matrix is plotted in Fig. 11. VLAD has produced a relatively good result of 26.4% and is comparable with IFK. The performance with different numbers of Gaussians is illustrated in Fig. 5.

The triangle assignment coding with max pooling gives the best result within

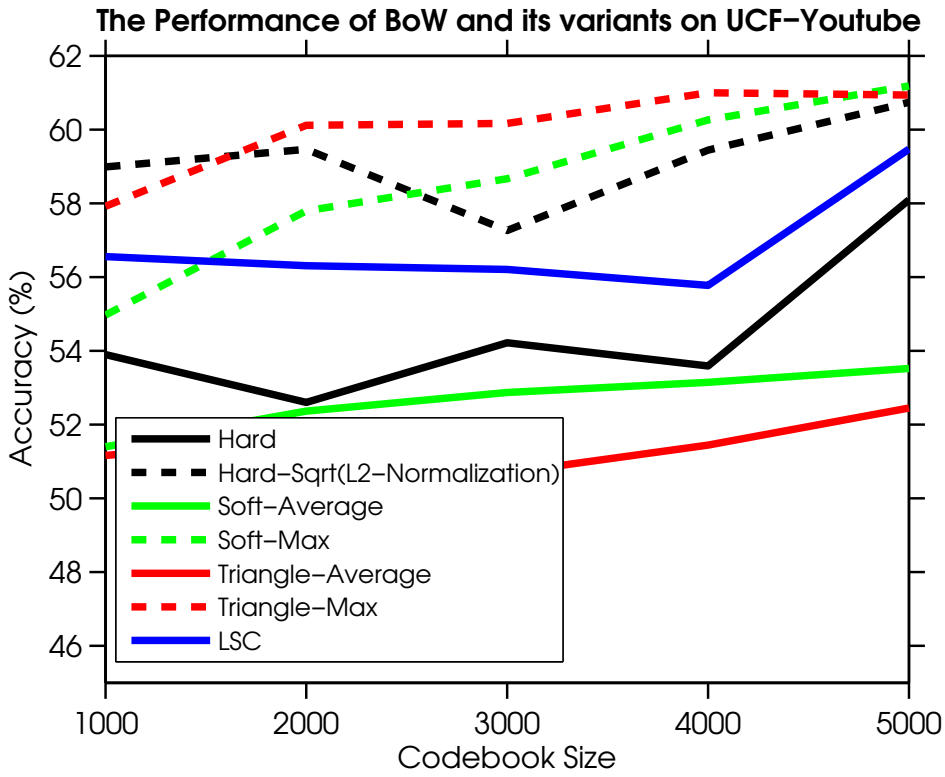

Figure 6: The performance of the BoW model and its variants on the UCF-YouTube dataset (recognition rate).

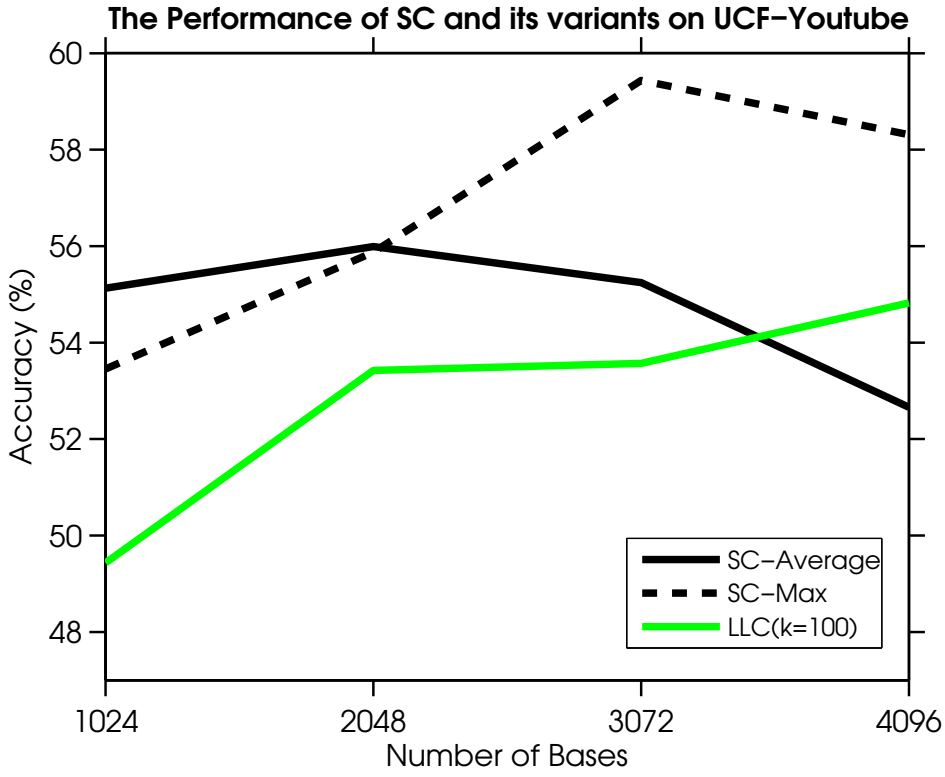

Figure 7: The performance of SC and its variants on the UCF-YouTube dataset (recognition rate).

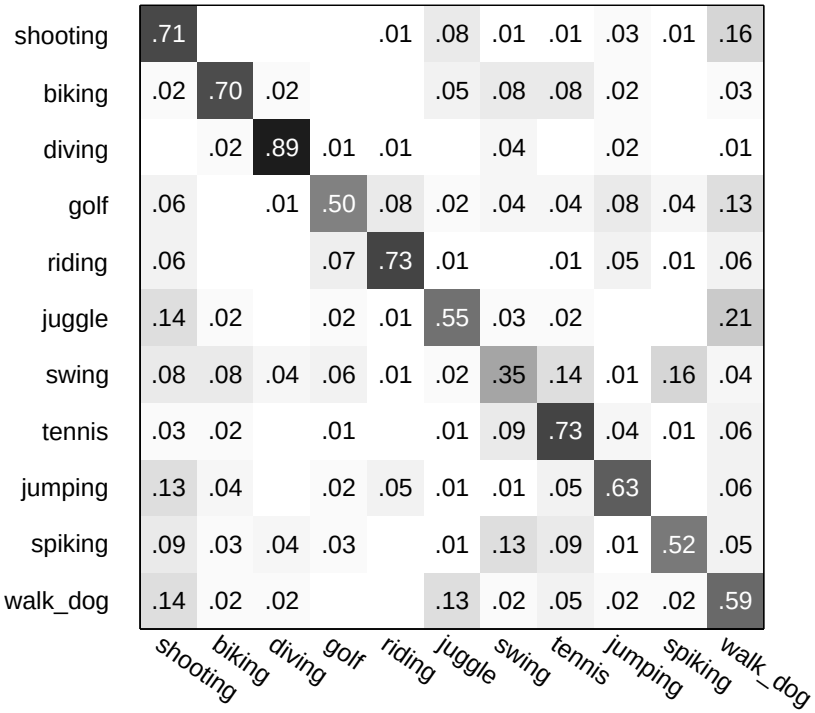

Figure 8: The confusion matrix of the best result with the NBNN kernel on the UCF Youtube dataset (recognition rate).

Kernels	Linear	RBF	χ^2	Intersection
BoW-Hard (Sqrt-L2-Normalization)	23.5%	17.2%	16.7%	22.0%
BoW-Soft-Max	24.0%	18.1%	-	-
BoW-Triangle-Max	25.1%	19.2%	-	-
BoW-LSC	24.6%	18.7%	-	-
SC-Max	27.9%	19.3%	-	-
LLC	24.1%	18.9%	-	-

Table 4: The performance of SVM with different kernels on HMDB51 (recognition rate).

the BoW model. LSC produces a comparable result of 24.6% with the triangle assignment coding. SC with max pooling achieves an impressive result -27.7%- which is better than all of the methods in BoWand SC. The performance of the NBNN family is similar to that on the UCF-YouTube dataset, where the NBNN kernel is better than both NBNN and local NBNN. The match kernel fails to provide reasonable results on this dataset.

Fig. 9 shows the performance of methods in BoW with different codebook sizes on the HMDB51 dataset. Most of the methods increase with codewords from 1000 to 5000, which is reasonable since this dataset is highly diverse with huge variations both intra and inter classes. As shown in Fig. 10, both SC and LLC become stable with the number of bases greater than 2048 with the best results around 3072. Similarly, the performance of SVM with different kernels on this dataset is illustrated in Figure 4. The intersection kernel is comparable with the linear kernel.

4.4. Discussions

In this section, we provide an in-depth discussion on the findings from experimental results and summarize the performance of different local methods which would be used a guidance for future research.

4.4.1. The BoW Model

The BoW model describes the probability distribution of local features by using voting-based histogram [23]. Each bin of the histogram represents the occurrence of a codeword in a video. However, it tends to be coarse and less informative due to quantization errors using a histogram, especially with a hard assignment. To compensate the information loss, many sophisticated coding methods have been developed. The newly proposed encoding techniques such as the triangle

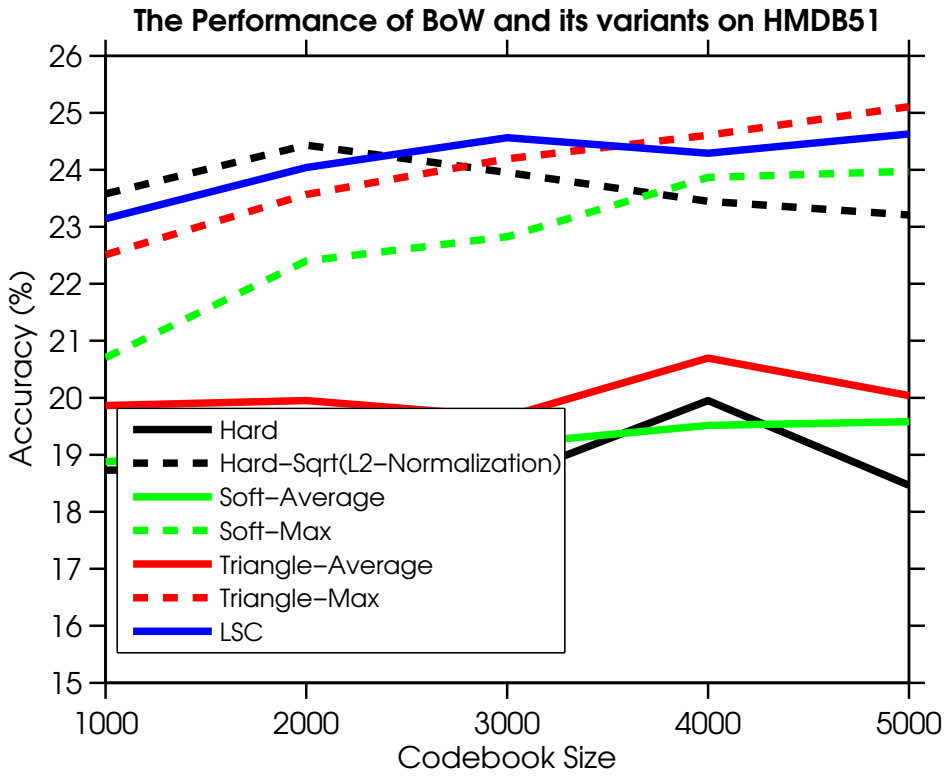

Figure 9: The performance of the BoW model and its variants on the HMDB51 dataset (recognition rate).

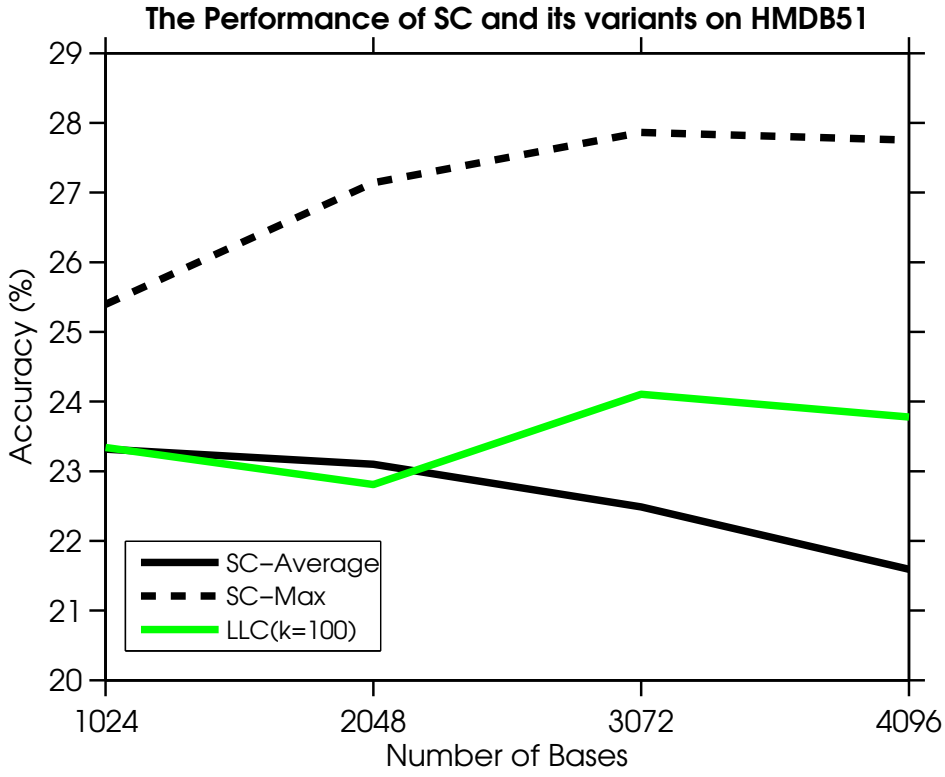

Figure 10: The performance of SC and its variants on the HMDB51 dataset (recognition rate).

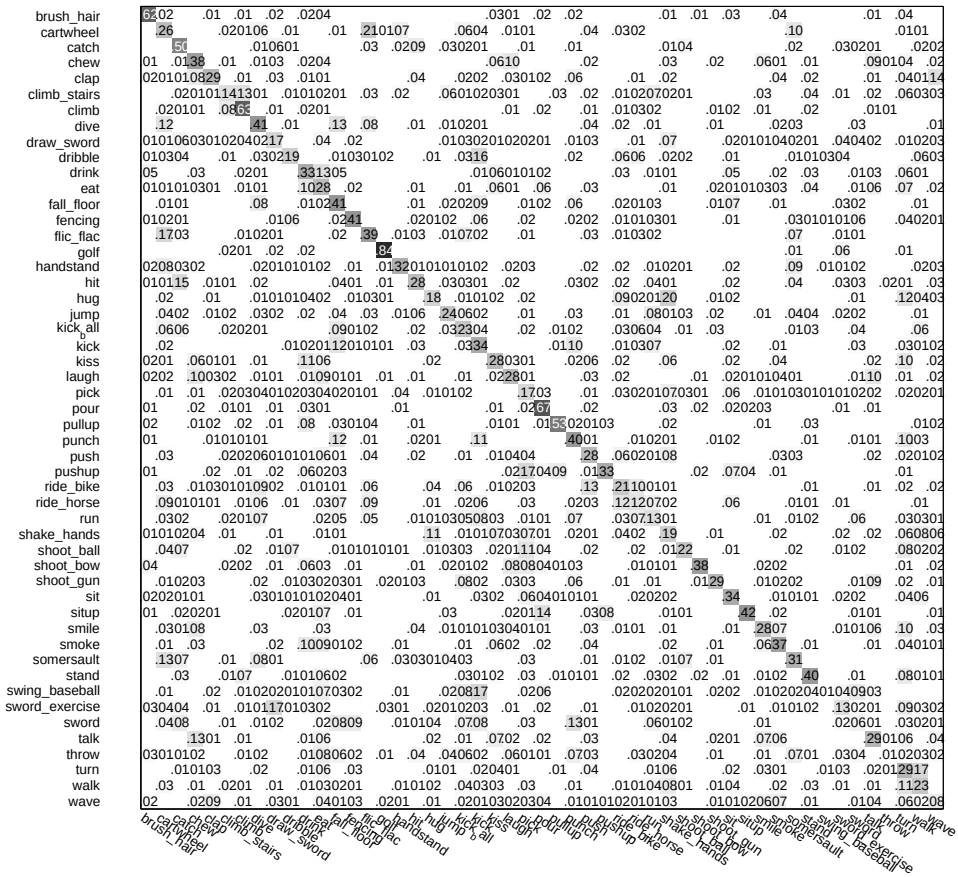

Figure 11: The confusion matrix of the result with the improved Fisher kernel on the HMDB51 dataset (recognition rate).

assignment coding with max pooling and the localized soft-assignment coding (LSC) significantly improve the baseline hard assignment coding, and achieve the state-of-the-art performance, especially on the KTH dataset. This is mainly because that the information loss during the feature quantization has been compensated by the sophisticated encoding techniques. The square root BoW with L2 normalization can further improve the regular BoW with hard assignment which is consistent with the findings in the image domain for object recognition [22]. Compared with other sophisticated encoding algorithms, *e.g.*, the IFK, the advantages of the BoW model lie in its theoretic simplicity and computational efficiency. It has been shown that the BoW is a special case of the Fisher kernel where the gradient computation is restricted to the mixture weight parameters of the GMM [71]. The BoW model with a hard assignment can be formulated in a match kernel framework with a linear kernel, which has been illustrated in [72].

4.4.2. Sparse Coding

Sparse coding (SC) is based on the reconstruction of local features with codewords by solving constrained least-square optimization problems. The obtained sparse codes of local features are pooled together to achieve a global representation of an action. Sparse coding and its variants via spatio-temporal local features have been extensively explored for action recognition [12]. With both average and max pooling strategies, SC outperforms most of the BoW based methods, which indicates its ability of encoding complex local features for action recognition. SC with max pooling has produced an impressive result of 27.9% on the HMDB51 dataset. The possible reason is that SC can better encode local features with less quantization errors while the max pooling makes it more insensitive to unusual local features. LLC does not outperform SC with max pooling on the three datasets, which is different from the performance on image classification tasks [16]. This is inconsistent with the report on object recognition in [16]. One reason could be that spatio-temporal features in video are much noisier than two-dimensional image features, which makes the locality constraint in LLC insignificant. In addition, LLC can produce reasonable results with more local neighbors k (over 100) than in the image domain (typically $k = 5$), which would be due to the fact that spatio-temporal local features in the video domain lie in a higher dimensional space. Therefore, to encode a local feature, more bases would be needed.

Note that, for all the methods using feature pooling, max pooling is significantly better than average pooling both in BoW and SC on the three datasets. This behavior is consistent with that in image classification [11]. Our experimental results have validated the effectiveness of max pooling for action recognition in the

video domain. Interestingly, the locality constraint and max pooling have demonstrated to be more effective in the BoW model than in SC, *e.g.*, LSC significantly improves the performance of BoW.

4.4.3. NBNN

Naive Bayes Nearest Neighbor (NBNN) is non-parametric approach first proposed for object classification and achieve state-of-the-art performance by avoidance of quantization errors in the BoW model. The NBNN family produces impressive results on all the three datasets, with highest recognition rate by the local NBNN classifier on the KTH dataset. Local NBNN generally outperforms NBNN on the three datasets. This is consistent with the results in image and object recognition [17, 18, 19]. However, the superiority of the NBNN family become less significant on more realistic datasets, *i.e.*, HMDB51, with a larger number of action categories. This would be due to that the assumption in NBNN that the smoothing parameter, namely the Parzen kernel bandwidth σ, is common for all categories does not, at least not fully, hold for large category numbers. Moroever, NBNN methods directly rely on the local features without mid-level feature encoding. On the realistic datasets, *e.g.*, HMDB51, local features are extremely noisy and therefore the performance of the NBNN family, *e.g.*, NBNN and local NBNN, would be seriously compromised since no training stage is used in NBNN and local NBNN. This can also explain that the NBNN kernel still shows good performance on realistic datasets because it employs a training stage which helps handle noisy features. Finally, the NBNN family is connected to the rest of local methods through the NBNN kernel which can also be formulated in terms of match kernels. Indeed, the local NBNN classifier can also be regarded as imposing the locality constraint on the original NBNN with max pooling if the distance to a neighbor is deemed as the inverse of similarity.

4.4.4. Fisher Kernel and VLAD

The Fisher kernel describes a video with a gradient vector derived from its probability function and the gradient vector indicates the directions in which parameters should be adjusted to fit the data [71]. The improved Fisher kernel (IFK) [21] has produced impressive results, especially on the realistic UCF-YouTube and challenging HMDB51 datasets. Compared to the BoW model, the FK is a more principled approach than the BoW to combine the generative and discriminative models. The Fisher vector encodes high-order statistics including the zeroth, first and second orders and describes how the set of descriptors deviates from an average distribution which is modeled by a parametric generative model [25].

Intuitively, the IFK has much higher dimensionality than the BoW model and therefore can encode much more information for representations which therefore produce better results.

The VLAD is a simplified non-probabilistic version of the FK [25] under the approximations that the soft assignment is replaced by a hard assignment and only the gradient with respect to the mean is considered [71]. The performance of vector of locally aggregated descriptor (VLAD) is competitive with the IFK while being more computational efficient and holds the same trends over all the datasets.

4.4.5. Match Kernels

Due to the simplicity, the match kernels yield relatively low recognition rates but sometimes are comparable to some of the methods in the BoW model such as the hard assignment, the soft and triangle assignments with average coding, especially on the KTH and HMDB51 datasets. With regard to match kernels, we have also experimented the max-sum kernel K_M, however, it performs far worse than the normalized sum kernel $K_{\mathcal{F}}$ and even fails to produce reasonable results on the UCF-YouTube dataset. This would be due to that it does not meet the Mercer condition and cannot guarantee that the optimization in SVM is convex as also shown in [73]. However, the most important role played by match kernels is the basic formulation of similarity of feature sets which can explain the connections among local feature based methods, including the BoW model [72], LLC [16] in sparse coding, the NBNN kernel [18], Fisher kernels [71] and VLAD [25].

4.4.6. Summary

To summarize, the IFK has shown superb performance for action recognition based on spatio-temporal local descriptors. This finding is consistent with that in image classification [23]. Although IFK does not always perform the best for the three datasets (slightly lower than local NBNN on KTH), its results on UCF-YouTube and HMDB51 are significantly better than other methods, showing the great potential of the IFK to handle complicated local features in realistic applications of human action recognition [23]. The VLAD has produced comparative performance, which is slightly lower than the IFK, showing significant advantages over the rest of the methods. The NBNN based methods have advantages on relatively simple datasets, *e.g.*, the KTH dataset, because of the innate avoidance of quantization errors by using image-to-class (I2C) distances. However, on realistic datasets, *e.g.*, UCF-YouTube and HMDB51, local features are extremely noisy, which makes the I2C distance less accurate and therefore the performance of NBNN based methods decreases. Although the BoW model and the sparse

coding algorithm have been widely used and shown their effectiveness for image classification and action recognition, their performance tend to be inferior on realistic datasets for action recognition even with sophisticated encoding methods. Match kernels yield inferior performance but provide a basic formulation that theoretically connects different local methods.

5. Conclusion

In this paper, we have done a comprehensive study on local methods for human action recognition. The state-of-the-art techniques, which have been widely used and shown effectiveness in the image domain, have been transferred to action recognition. Extensive experiments have been conducted to systematically evaluate and compare these techniques on three benchmark datasets: KTH, UCF-YouTube and HMDB51. Moreover, we have also provided experimental and theoretical insights into the performance of each method and drawn useful conclusions from findings in the experiments. As many of the techniques are innovated in the image domain and have not yet been applied to action recognition, our work can serve as guidance for future research in action recognition.

References

[1] J. Sivic, A. Zisserman, Video google: A text retrieval approach to object matching in videos, in: IEEE International Conference on Computer Vision, 2003, pp. 1470–1477.

[2] A. Kläser, M. Marszałek, C. Schmid, A spatio-temporal descriptor based on 3d-gradients, in: British Conference on Computer Vision, 2008, pp. 995–1004.

[3] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[4] I. Laptev, T. Lindeberg, Space-time interest points, in: IEEE International Conference on Computer Vision, 2003.

[5] T. de Campos, M. Barnard, K. Mikolajczyk, J. Kittler, F. Yan, W. Christmas, D. Windridge, An evaluation of bags-of-words and spatio-temporal shapes for action recognition, in: Winter Application and Computer Vision, 2011, pp. 344–351.

[6] S.-F. Wong, R. Cipolla, Extracting spatiotemporal interest points using global information, in: IEEE 11th International Conference on Computer Vision, 2007, pp. 1–8.

[7] A. Oikonomopoulos, I. Patras, M. Pantic, Spatiotemporal salient points for visual recognition of human actions, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 36 (3) (2005) 710–719.

[8] X. Zhen, L. Shao, D. Tao, X. Li, Embedding motion and structure features for action recognition, IEEE Transactions on Circuits and System for Video Technoloy 23 (7) (2013) 1182–1190.

[9] L. Shao, X. Zhen, D. Tao, X. Li, Spatio-temporal laplacian pyramid coding for action recognition, IEEE Transactions on Cybernetics 44 (6) (2014) 817–827.

[10] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1794–1801.

[11] Y. Boureau, F. Bach, Y. LeCun, J. Ponce, Learning mid-level features for recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 2559–2566.

[12] T. Guha, R. K. Ward, Learning sparse representations for human action recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (8) (2012) 1576–1588.

[13] J. van Gemert, C. Veenman, A. Smeulders, J. Geusebroek, Visual word ambiguity, IEEE Transactions on Pattern Analysis and Artifical Intelligence 32 (7) (2010) 1271–1283.

[14] A. Coates, H. Lee, A. Ng, An analysis of single-layer networks in unsupervised feature learning, Ann Arbor 1001 (2010) 48109.

[15] J. Liu, J. Luo, M. Shah, Recognizing realistic actions from videos in the wild, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1996–2003.

[16] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 3360–3367.

[17] O. Boiman, E. Shechtman, M. Irani, In defense of nearest-neighbor based image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[18] T. Tuytelaars, M. Fritz, K. Saenko, T. Darrell, The nbnn kernel, in: IEEE International Conference on Computer Vision, 2011, pp. 1824–1831.

[19] S. McCann, D. Lowe, Local naive bayes nearest neighbor for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3650–3656.

[20] F. Perronnin, C. Dance, Fisher kernels on visual vocabularies for image categorization, in: IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.

[21] F. Perronnin, J. Sánchez, T. Mensink, Improving the fisher kernel for large-scale image classification, in: European Conference on Computer Vision, Springer, 2010, pp. 143–156.

[22] R. G. Cinbis, J. Verbeek, C. Schmid, Image categorization using fisher kernels of non-iid image models, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2184–2191.

[23] Y. Huang, Z. Wu, L. Wang, T. Tan, Feature coding in image classification: A comprehensive study, IEEE Transactions on Pattern Analysis and Artifical Intelligence 36 (3) (2013) 493–506.

[24] H. Jégou, M. Douze, C. Schmid, P. Pérez, Aggregating local descriptors into a compact image representation, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 3304–3311.

[25] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, C. Schmid, Aggregating local image descriptors into compact codes, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (9) (2012) 1704–1716.

[26] S. Lyu, Mercer kernels for object recognition with local features, in: IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, 2005, pp. 223–229.

[27] C. Wallraven, B. Caputo, A. Graf, Recognition with local features: the kernel recipe, in: IEEE International Conference on Computer Vision, 2003, pp. 257–264.

[28] H. Wang, M. Ullah, A. Klaser, I. Laptev, C. Schmid, et al., Evaluation of local spatio-temporal features for action recognition, in: British Conference on Computer Vision, 2009.

[29] L. Shao, R. Mattivi, Feature detector and descriptor evaluation in human action recognition, in: ACM International Conference on Image and Video Retrieval, 2010, pp. 477–484.

[30] A. Tamrakar, S. Ali, Q. Yu, J. Liu, O. Javed, A. Divakaran, H. Cheng, H. Sawhney, Evaluation of low-level features and their combinations for complex event detection in open source videos, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3681–3688.

[31] I. Everts, J. C. van Gemert, T. Gevers, Evaluation of color stips for human action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2850–2857.

[32] K. K. Reddy, M. Shah, Recognizing 50 human action categories of web videos, Machine Vision and Applications 1–11.

[33] U. Gaur, Y. Zhu, B. Song, A. Roy-Chowdhury, A string of feature graphs model for recognition of complex activities in natural videos, in: IEEE International Conference on Computer Vision, 2011, pp. 2595–2602.

[34] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, J. Li, Hierarchical spatio-temporal context modeling for action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 2004–2011.

[35] H. Wang, A. Kläser, C. Schmid, L. Cheng-Lin, Action recognition by dense trajectories, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2011.

[36] P. Matikainen, M. Hebert, R. Sukthankar, Representing pairwise spatial and temporal relations for action recognition, 2010, pp. 508–521.

[37] T. H. Thi, L. Cheng, J. Zhang, L. Wang, S. Satoh, Structured learning of local features for human action classification and localization, Image and Vision Computing 30 (1) (2012) 1–14.

[38] J. Wang, Z. Chen, Y. Wu, Action recognition with multiscale spatio-temporal contexts, in: IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 3185–3192.

[39] C. Yuan, X. Li, W. Hu, H. Ling, S. Maybank, 3d r transform on spatio-temporal interest points for action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013.

[40] A. Kovashka, K. Grauman, Learning a hierarchy of discriminative space-time neighborhood features for human action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 2046–2053.

[41] Y. Zhang, X. Liu, M.-C. Chang, W. Ge, T. Chen, Spatio-temporal phrases for activity recognition, in: European Conference on Computer Vision, 2012, pp. 707–721.

[42] Z. Lu, Y. Peng, H. H. Ip, Spectral learning of latent semantics for action recognition, in: IEEE International Conference on Computer Vision, 2011, pp. 1503–1510.

[43] S. Wang, Y. Yang, Z. Ma, X. Li, C. Pang, A. G. Hauptmann, Action recognition by exploring data distribution and feature correlation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1370–1377.

[44] Q. V. Le, W. Y. Zou, S. Y. Yeung, A. Y. Ng, Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, 2011, pp. 3361–3368.

[45] M. Jain, H. Jégou, P. Bouthemy, et al., Better exploiting motion for better action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013.

[46] N. Shapovalova, A. Vahdat, K. Cannons, T. Lan, G. Mori, Similarity constrained latent support vector machine: an application to weakly supervised action classification, in: European Conference on Computer Vision, Springer, 2012, pp. 55–68.

[47] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, C.-W. Ngo, Trajectory-based modeling of human actions with motion reference points, in: Europen Conference on Computer Vision, 2012.

[48] T. Dean, G. Corrado, R. Washington, Sparse spatiotemporal coding for activity recognition, Tech. rep. (2010).

[49] Y. Zhu, X. Zhao, Y. Fu, Y. Liu, Sparse coding on local spatial-temporal volumes for human action recognition, in: Asian Conference on Computer Vision, Springer, 2011, pp. 660–671.

[50] D. Oneata, J. Verbeek, C. Schmid, Action and event recognition with fisher vectors on a compact feature set, in: Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE, 2013, pp. 1817–1824.

[51] V. Kantorov, I. Laptev, Efficient feature extraction, encoding, and classification for action recognition, in: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE, 2014, pp. 2593–2600.

[52] X. Peng, C. Zou, Y. Qiao, Q. Peng, Action recognition with stacked fisher vectors, in: Computer Vision–ECCV 2014, Springer, 2014, pp. 581–595.

[53] E. Vig, M. Dorr, D. Cox, Space-variant descriptor sampling for action recognition based on saliency and eye movements, in: European Conference on Computer Vision, 2012, pp. 84–97.

[54] Z. Cai, L. Wang, X. Peng, Y. Qiao, Multi-view super vector for action recognition, in: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE, 2014, pp. 596–603.

[55] B. Wu, C. Yuan, W. Hu, Human action recognition based on context-dependent graph kernels, in: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE, 2014, pp. 2609–2616.

[56] X. Yang, Y. Tian, Action recognition using super sparse coding vector with spatio-temporal awareness, in: Computer Vision–ECCV 2014, Springer, 2014, pp. 727–741.

[57] L. Sun, K. Jia, T.-H. Chan, Y. Fang, G. Wang, S. Yan, Dl-sfa: Deeply-learned slow feature analysis for action recognition, in: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE, 2014, pp. 2625–2632.

[58] Z. Lan, M. Lin, X. Li, A. G. Hauptmann, B. Raj, Beyond gaussian pyramid: Multi-skip feature stacking for action recognition, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[59] S. Yang, C. Yuan, B. Wu, W. Hu, F. Wang, Multi-feature max-margin hierarchical bayesian model for action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1610–1618.

[60] A. Vedaldi, B. Fulkerson, Vlfeat: An open and portable library of computer vision algorithms, in: ACM International Conference on Multimedia, 2010, pp. 1469–1472.

[61] L. Liu, L. Wang, X. Liu, In defense of soft-assignment coding, in: IEEE International Conference on Computer Vision, 2011, pp. 2486–2493.

[62] J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online dictionary learning for sparse coding, in: International Conference on Machine Learning, 2009, pp. 689–696.

[63] T. Jaakkola, D. Haussler, et al., Exploiting generative models in discriminative classifiers, Advances in neural information processing systems (1999) 487–493.

[64] I. Laptev, B. Caputo, C. Schüldt, T. Lindeberg, Local velocity-adapted motion events for spatio-temporal recognition, Computer Vision and Image Understanding 108 (3) (2007) 207–229.

[65] C. Schuldt, I. Laptev, B. Caputo, Recognizing human actions: A local svm approach, in: International Conference on Pattern Recognition, Vol. 3, 2004, pp. 32–36.

[66] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre, Hmdb: A large video database for human motion recognition, in: IEEE International Conference on Computer Vision, 2011, pp. 2556–2563.

[67] P. Dollár, V. Rabaud, G. Cottrell, 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, pp. 65–72.

[68] J. Liu, M. Shah, Learning human actions via information maximization, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[69] C.-C. Chang, C.-J. Lin, Libsvm: a library for support vector machines, ACM Transactions on Intelligent Systems and Technology 2 (3) (2011) 27.

[70] H. Wang, A. Kläser, C. Schmid, C.-L. Liu, Dense trajectories and motion boundary descriptors for action recognition, International Journal of Computer Vision 103 (1) (2013) 60–79.

[71] J. Sánchez, F. Perronnin, T. Mensink, J. Verbeek, Image classification with the fisher vector: Theory and practice, International journal of computer vision 105 (3) (2013) 222–245.

[72] L. Bo, C. Sminchisescu, Efficient match kernel between sets of features for visual recognition, Vol. 2, 2009.

[73] B. Caputo, L. Jie, A performance evaluation of exact and approximate match kernels for object recognition, Electronic Letters on Computer Vision and Image Analysis 8 (3) (2009) 15–26.