A Performance Evaluation on Action Recognition with Local Features

Xiantong Zhen
Department of Electronic and Electrical Engineering
The University of Sheffiled
Email: elr10xz@sheffield.ac.uk

Ling Shao
Department of Electronic and Electrical Engineering
The University of Sheffiled
Email: ling.shao@sheffield.ac.uk

***Abstract*—Local features have played an important role in visual recognition. Methods based on local features, e.g., the bag-of-words (BoW) model and sparse coding, have shown their effectiveness in image and object recognition in the past decades. Recently, many new techniques, including the improvements of BoW and sparse coding as well as the non-parametric naive bayes nearest neighbor (NBNN) classifier, have been proposed and advanced the state-of-the-art in the image domain.**

However, in the video domain, the BoW model still dominates the action recognition field. It is unclear how effective the state-of-the-art techniques widely used in the image domain would perform on action recognition. To fill this gap, we aim to implement and provide a systematic study of these techniques on action recognition, and compare their performance under a unified evaluation framework. Other techniques such as match kernels and random forest, which have also demonstrated their potential in handling local features, are also included for a comprehensive evaluation. Extensive experiments have been conducted on three benchmarks including the KTH, the UCF-YouTube and the HMDB51 datasets, and results and findings are analyzed and discussed

I. Introduction

Human action recognition has been an active topic in the computer vision community for many years. Most of the current methods, from low-level feature extraction to high-level feature representations, in action recognition are extended from the text and image domains. Local features have shown increasing effectiveness in visual recognition, and local methods based on spatio-temporal local features, e.g., HOG3D [11], become popular in action recognition since the inventions of spatio-temporal interest points detectors [9], [14]. In contrast to holistic representations [38], [26], local methods enjoy many advantages such as resistance to occlusion and clutter. Moreover, localization, bounding boxes or tracking, which are intractable in realistic scenarios, are not required.

Methods using sparsely detected local features, e.g., the bag-of-word (BoW) model and sparse coding (SC), have obtained remarkable performance in image and object classification. Recently, refinements of BoW and SC as well as alternative techniques 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.

Since the introduction of the BoW model to video analysis [27], it has dominated in action recognition due to its conceptual simplicity and computational efficiency. However, the BoW model is criticized because quantization errors are incurred during its creation and the errors would be propagated to the final representation. To alleviate these deficits, various coding algorithms based on the BoW model have been introduced, including the soft assignment coding (kernel codebooks) [31], and localized soft-assignment coding (LSC) [15].

Recently, sparse coding [36], [35], [3] based on local features such as SIFT [18] has been successfully used for image representations. It inherently has the nature of alleviating quantization errors incurred in BoW by encoding local features with multiple bases. By incorporating the locality into the coding process, the locality-constrained linear coding (LLC) [35], outperforms the ordinary sparse coding in image and object classification.

Match kernels between sets of local features have long been exploited in visual recognition [19], [33]. Match kernels are able to compute the similarity between sets of unordered local features and have achieved state-of-the-art results in image and object recognition. Actually, the BoW model can be considered as a special version of match kernels [1].

Instead of explicitly representing images by coding local features, a simple non-parametric nearest neighbor (NN) based classifier, naive bayes nearest neighbor (NBNN), was proposed in [2]. By computing the 'Image-to-Class' rather than 'Image-to-Image' distance, NBNN is able to avoid quantizing the 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 [29] and the local NBNN [21], have also been developed. The NBNN family have shown excellent effectiveness in image and object recognition.

The above-mentioned techniques have been widely used and demonstrated the 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 common experimental settings.

Methods using tracking of trajectories can always outperform those based on STIPs while requiring higher computational complexity [10]. What's more, Reddy and Shah [23] found that 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 [34], [25], and apply the same feature detection and description steps to all the methods to be evaluated. Thorough comparisons are carried out on three typical action datasets, *i.e.*, KTH, UCF-YouTube and HMDB51. In addition, we provide detailed analysis and draw impartial conclusions from the findings in the experiments.

A. Related work

Performance evaluations have gained increasing attention in computer vision with the large number and variety of algorithms being developed. Plenty of evaluation and analysis works have been conducted both in the image domain [22], [37], [4], [30], [7], [6] and on action recognition [34], [25], [8], [28].

A recent work in [7] closely related to ours investigated the performance of unsupervised feature learning algorithms with single-layer networks on image classification. Surprisingly, the best performance from their evaluation is obtained by the BoW model with the so called triangle assignment coding. In addition, Chatifeld *et al.* [6] presented a comprehensive evaluation and deep analysis of the feature encoding methods within the BoW model for image classification.

Two important evaluation works on action recognition were conducted by Wang *et al.* [34] and Shao and Mattivi [25]. They evaluated and compared the performance of different detectors and descriptors as well as their combinations for action recognition. However, both of them used only the standard BoW model for action representation with a support vector machine (SVM) classifier.

Campos *et al.* [4] have compared the BoW model with spatio-temporal shapes (STS) for action recognition. Two versions of the BoW-based methods, namely spatially-constrained BoW (SBoW) and local-BoW (LBoW), were considered. The 3-dimonsional histogram of oriented gradients (HOG3D) [11] was employed as the spatio-temporal descriptor.

Ramrakar *et al.* [28] evaluated low-level features and their combinations for complex event detection. Extensive low-level features, including static visual features and dynamic visual features, are adopted for comparison. Again, the BoW model has been utilized as the final representation in their work.

Recently, Everts et al. [10] have done an evaluation on color STIPs for human action recognition. By incorporating the chromatic representations into the spatio-temporal domain, they reformulated the STIP detectors and descriptors for multi-channel video representation, which are shown to outperform the intensity-based counterparts.

The above evaluations were either in the image domain or centered on the BoW model. Our paper is, however, concentrated on the evaluation of state-of-the-art techniques on action recognition in the video domain.

B. Overview

In Section 2, we revisit the state-of-the-art methods based on local features. In Section 3, we describe the implementation details of each method when they are applied to action recognition and provide the experimental results and discussions. Finally, we conclude this work in Section 4.

II. Methods

In this Section, we describe the widely used methods based on local features for visual recognition.

A. The Bag-of-Words (BoW) model

Local features in the training set are first clustered to create a codebook [32]. 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 [31], the triangle assignment [7] and the localized soft assignment [17].

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 $B_{D\times M}$ 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 $\mathbf{b}_j$.

1) Hard assignment coding: 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

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

2) Soft assignment coding: In the soft assignment coding, The coefficient $u_{i,j}$ is the degree of membership of a local feature $\mathbf{x}_i$ to the jth codeword.

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

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

3) Triangle assignment coding: The triangle assignment coding was proposed in [7]. The coding is defined by the following activation function:

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

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.

4) Localized soft assignment coding (LSC): By combining the ideas of localization and the soft assignment coding, Liu *et al.* [17] proposed the localized soft-assignment coding (LSC). The activation function takes the form in Eq. (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 $d(\mathbf{x}_i, \mathbf{b}_j)$.

B. 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 [20]:

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

where λ controls the sparsity of the coefficient.

1) Locality-constrained linear coding (LLC): Instead of enforcing sparsity in SC, LLC [35] 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 $dist(\mathbf{x}_i, \mathbf{B}) = [dist(\mathbf{x}_i, \mathbf{b}_1), ..., dist(\mathbf{x}_i, \mathbf{b}_M)]^{\mathrm{T}}$, 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.

C. Match kernels

Match kernels between sets of local features have long been exploited [33], [19]. 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:

$$K_S(\mathcal{F}_a, \mathcal{F}_b) = \frac{1}{|\mathcal{F}_a|} \frac{1}{|\mathcal{F}_b|} \sum_{i=1}^{|\mathcal{F}_a|} \sum_{j=1}^{|\mathcal{F}_b|} K_F(F_i^{(a)}, F_j^{(b)}) \tag{8}$$

In [33], a kernel function (the max-sum kernel) for matching local features was proposed:

$$\begin{aligned} K_M(\mathcal{F}_a, \mathcal{F}_b) = \frac{1}{2} \sum_{i=1}^{|\mathcal{F}_a|} \max_{j=1,\ldots,|\mathcal{F}_b|} K_F(F_i^{(a)}, F_j^{(b)}) \\ + \frac{1}{2} \sum_{j=1}^{|\mathcal{F}_b|} \max_{i=1,\ldots,|\mathcal{F}_a|} K_F(F_j^{(b)}, F_i^{(a)}) \end{aligned} \tag{9}$$

This match kernel has been used in objection recognition [33] and action classification [13]. Lyu *et al.* [19] 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:

$$K_{\mathcal{F}}(\mathcal{F}_a, \mathcal{F}_b) = \frac{1}{|\mathcal{F}_a|} \frac{1}{|\mathcal{F}_b|} \sum_{i=1}^{|\mathcal{F}_a|} \sum_{j=1}^{|\mathcal{F}_b|} [K_F(F_i^{(a)}, F_j^{(b)})]^p, \tag{10}$$

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

D. 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:

$$\hat{C} = arg\max_C p(C|Q) = arg\max_C p(Q|C). \tag{11}$$

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

$$p(Q|C) = p(\mathbf{x}_1, \ldots, \mathbf{x}_N|C) = \prod_{i=i}^{N} p(\mathbf{x}_i|C) \tag{12}$$

$p(\mathbf{x}_i|C)$ 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:

$$\bar{c} = arg\min_c = \sum_{\mathbf{x} \in \mathbf{X}} ||\mathbf{x} - NN^c(\mathbf{x})||^2, \tag{13}$$

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

1) NBNN kernel: The NBNN kernel is based on the normalized sum match kernel [19], to calculate the similarity between two sets of features $X = \{\mathbf{x}\}$ and $Y = \{\mathbf{y}\}$:

$$\begin{aligned} K(X, Y) &= \sum_{c \in C} K^c(X, Y) \\ &= \frac{1}{|X||Y|} \sum_{c \in C} \sum_{\mathbf{x} \in X} \sum_{\mathbf{y} \in Y} k^c(\mathbf{x}, \mathbf{y}), \end{aligned} \tag{14}$$

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:

$$k^c(\mathbf{x},\mathbf{y}) = \phi^c(\mathbf{x})^{\mathrm{T}}\phi^c(\mathbf{y}) = f^c(d_{\mathbf{x}}^1,\ldots,d_{\mathbf{x}}^{|C|})^{\mathrm{T}} f^c(d_{\mathbf{y}}^1,\ldots,d_{\mathbf{y}}^{|C|}) \quad (15)$$

Two distance functions have been considered in the original work [29], namely,

$$f_1^c(d_{\mathbf{x}}^1,\ldots,d_{\mathbf{x}}^{|C|}) = d_{\mathbf{x}}^c,$$
$$f_2^c(d_{\mathbf{x}}^1,\ldots,d_{\mathbf{x}}^{|C|}) = d_{\mathbf{x}}^c - d_{\mathbf{x}}^{\hat{c}}, \quad (16)$$

where $d_{\mathbf{x}}^c$ is the distance to its nearest neighbor in class c and $d_{\mathbf{x}}^{\hat{c}}$ denotes the closest distance to all classes except for c.

2) Local NBNN: McCann and Lowe [21] 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.

III. Experiments and Results

A. Datasets

The KTH dataset [24] is a commonly used benchmark action dataset with 2391 video clips and six human action classes performed by 25 subjects. We follow the standard experimental setup [34], *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] is challenging due to large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background and illumination condition. This dataset contains a total of 1168 sequences with 11 action categories. We follow the experimental settings in [15].

The HMDB51 dataset [12] 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. All the results are reported by averaging the three training/test splits [12].

B. Experimental settings

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

Spatio-temporal local features. We employ the periodic detector proposed by Dollar at al. [9] to detect the spatio-temporal interest points from the raw video sequences and follow the parameter settings in the evaluation work of [34]. As in [7], the three-dimensional histogram of oriented gradients (HOG3D) [11] is used to describe each STIP due to its computational efficiency. The chosen detector and descriptor have shown outstanding performance in [34], [25]. 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) [16] 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.

Feature pooling. In BoW and SC, a final representation $\mathbf{P} \in R^M$ of an action is obtained by pooling over the coefficients [3]. With average pooling, the jth component of $\mathbf{P}$ is obtained by $p_j = \sum_{i=1}^{N} u_{ij}/N$. With max pooling, p_j is obtained by $p_j = \max_i u_{ij}$, where $i = 1, 2, \cdots, N$.

The BoW model. In the BoW model, the codebooks are created by the k-means clustering algorithm provided in VLFeat toolbox [32]. In LSC, we follow the parameter settings in the original work [17] with β in Eq. (2) set as 10.

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 [20], and the sparse codes are learned using orthogonal matching pursuit (OMP) [20]. The parameter λ in Eq. (5) 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.

Naive bayes nearest neighbors (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.

Match kernels. For the match kernels, we use the linear kernel as the local kernel and the single parameter p in Eq. (10) is set as 9 according to the original work [19]. We also use the normalized kernel in building the SVM classifier: $K(x,y) \leftarrow \frac{K(x,y)}{\sqrt{K(x,x)}\sqrt{K(y,y)}}$.

Action classification. We use a support vector machine (SVM) [5] classifier for BoW, SC and the match kernels. Note that a linear kernel instead of the χ^2 kernel in [34] is used in BoW and SC to make fair comparisons.

C. Results

All the final results on the three datasets are shown in Table 1. The size of the codebook in BoW and the number of bases in SC are hard to pre-determine while always affect the performance. Therefore, we have investigated the effects and illustrated the results in Fig. 1 and Fig, 2, respectively.

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 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

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

TABLE I. THE PERFORMANCE OF ALL METHODS ON THREE DATASETS, *i.e.*, KTH, UCF-YOUTUBE AND HMDB51. NOTE THAT THE RESULTS OF THE MATCH KERNEL ARE OBTAINED BY $K_{\mathcal{F}}$.

Methods	KTH	YouTube	HMDB
BoW-Hard	87.9%	58.1%	20.0%
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%
Match Kernel	86.9%	54.5%	13.7%

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 [7]. Note that our implementation of the baseline hard assignment coding is lower than that in [34], which would be due to that a χ^2 kernel is employed in their work. The ordinary SC with max pooling achieves even better results than LLC.

2) On the UCF-YouTube dataset: The results on the UCF-YouTube dataset are similar to those on the KTH dataset. The NBNN kernel produces the best result of 62.4%. Differently, the soft assignment coding with max pooling beats LSC and achieves the best result of 62.1% within the BoW family. In addition, SC with max pooling outperforms LLC obtaining an accuracy of 59.4%.

3) On the HMDB51 dataset: Slightly different on the HMDB51 dataset, the sparse coding models demonstrate relatively good performance. The best result -27.9%- is obtained by SC with max pooling. The triangle assignment coding gives the best result with the BoW model. LSC produces a comparable result -24.6%- with the triangle assignment coding -25.1%- which is the best in BoW. In addition, the NBNN kernel produces an impressive result on this dataset.

D. Summary and Discussion

The NBNN family produce remarkable results on all the three datasets, with highest recognition rates by the local NBNN classifier on KTH and by the NBNN kernel on UCF-YouTube. This is consistent with the results in image and object recognition [2], [29], [21]. However, we can see from Table 1 that 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.

The number of k the nearest neighbor in local NBNN is the only parameter. We have also evaluated the effects of k on local NBNN, which, however, only slightly affects the performance with the k ranging from 5 to 30 in our experiments.

Although the BoW model has long been criticized for its quantization errors, the newly proposed techniques such as the triangle 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 KTH. This is mainly because that the information loss during the feature quantization has been compensated by the sophisticated coding techniques and of the powerful classifier, *i.e.*, SVMs.

With both average and max pooling, SC outperforms most of the BoW based methods, which indicates its potential on action recognition. However, LLC does not outperform SC with max pooling on the three datasets. This is inconsistent with the report on object recognition in [35]. One reason could be that spatio-temporal features in video are much noisier than 2D features, which makes the locality constraint in LLC insignificant.

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 [3].

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

Finally, the recognition rates of the match kernels are relatively low but are comparable to some of the methods in the BoW model such as the hard assignment, the soft and triangle assignments with average coding on KTH and UCF-Sports.

IV. CONCLUSION

In this paper, we have transferred the state-of-the-art techniques, which have been widely used and shown effectiveness in the image domain, 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.

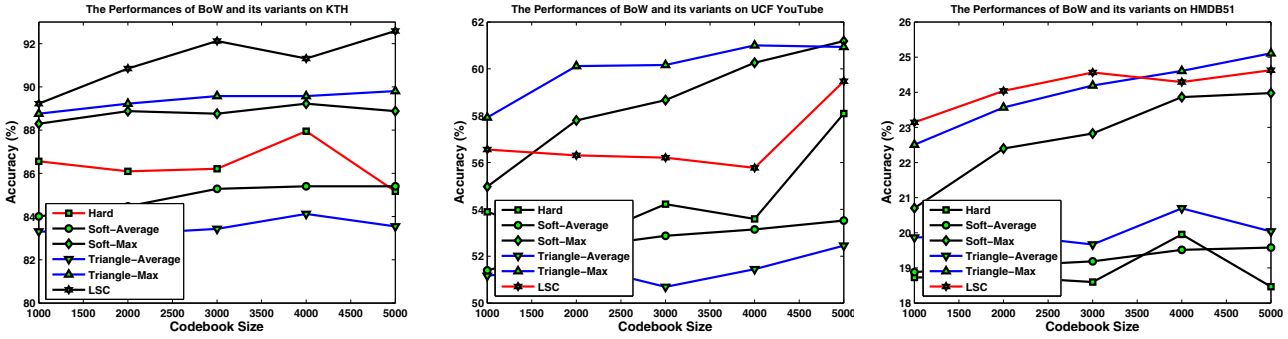
REFERENCES

[1] L. Bo and C. Sminchisescu. Efficient match kernel between sets of features for visual recognition. *NIPS*, 2(3), 2009.

[2] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *CVPR*, pages 1–8, 2008.

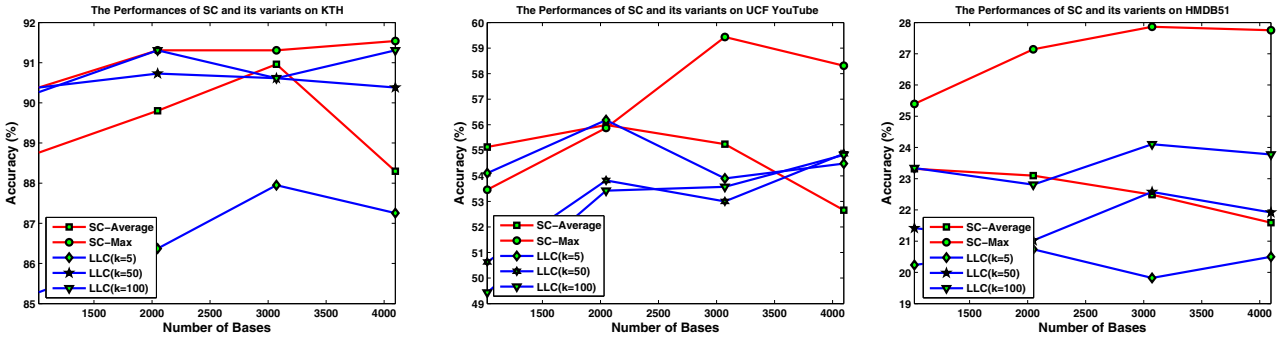
[3] Y. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *CVPR*, pages 2559–2566, 2010.

[4] B. Caputo and L. Jie. A performance evaluation of exact and approximate match kernels for object recognition. *ELCVIA*, 8(3):15–26, 2009.



(a) (b) (c)

Fig.1. Performance of the BoW model and its variants with different sizes of codebooks.



(a) (b) (c)

Fig.2. Performance of sparse coding and its variant with different sizes of dictionaries.

[5] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM-TIST*, 2:27:1–27:27, 2011.

[6] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, 2011.

[7] A. Coates, H. Lee, and A. Ng. An analysis of single-layer networks in unsupervised feature learning. *Ann Arbor*, 1001:48109, 2010.

[8] T. de Campos, M. Barnard, K. Mikolajczyk, J. Kittler, F. Yan, W. Christmas, and D. Windridge. An evaluation of bags-of-words and spatio-temporal shapes for action recognition. In *WACV*, pages 344–351, 2011.

[9] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, pages 65–72, 2005.

[10] I. Everts, J. C. van Gemert, and T. Gevers. Evaluation of color stips for human action recognition. 2013.

[11] A. Kläser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, pages 995–1004, sep 2008.

[12] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: A large video database for human motion recognition. In *ICCV*, pages 2556–2563, 2011.

[13] I. Laptev, B. Caputo, C. Schüldt, and T. Lindeberg. Local velocity-adapted motion events for spatio-temporal recognition. *CVIU*, 108(3):207–229, 2007.

[14] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, 2003.

[15] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos in the wild. In *CVPR*, pages 1996–2003, 2009.

[16] J. Liu and M. Shah. Learning human actions via information maximization. In *CVPR*, pages 1–8, 2008.

[17] L. Liu, L. Wang, and X. Liu. In defense of soft-assignment coding. In *ICCV*, pages 2486–2493, 2011.

[18] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[19] S. Lyu. Mercer kernels for object recognition with local features. In *CVPR*, volume 2, pages 223–229, 2005.

[20] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *ICML*, pages 689–696, 2009.

[21] S. McCann and D. Lowe. Local naive bayes nearest neighbor for image classification. In *CVPR*, pages 3650–3656, 2012.

[22] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *TPAMI*, 27(10):1615–1630, 2005.

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

[24] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *ICPR*, volume 3, pages 32–36, 2004.

[25] L. Shao and R. Mattivi. Feature detector and descriptor evaluation in human action recognition. In *CIVR*, pages 477–484, 2010.

[26] L. Shao, X. Zhen, D. Tao, and X. Li. Spatio-temporal laplacian pyramid coding for action recognition. *IEEE TCYB*, 2013.

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

[28] A. Tamrakar, S. Ali, Q. Yu, J. Liu, O. Javed, A. Divakaran, H. Cheng, and H. Sawhney. Evaluation of low-level features and their combinations for complex event detection in open source videos. In *CVPR*, pages 3681–3688, 2012.

[29] T. Tuytelaars, M. Fritz, K. Saenko, and T. Darrell. The nbnn kernel. In *ICCV*, pages 1824–1831, 2011.

[30] K. Van De Sande, T. Gevers, and C. Snoek. Evaluating color descriptors for object and scene recognition. *TPAMI*, 32(9):1582–1596, 2010.

[31] J. van Gemert, C. Veenman, A. Smeulders, and J. Geusebroek. Visual word ambiguity. *TPAMI*, 32(7):1271–1283, 2010.

[32] A. Vedaldi and B. Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. In *ACM-MM*, pages 1469–1472, 2010.

[33] C. Wallraven, B. Caputo, and A. Graf. Recognition with local features: the kernel recipe. In *ICCV*, pages 257–264, 2003.

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

[35] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, pages 3360–3367, 2010.

[36] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, pages 1794–1801, 2009.

[37] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *IJCV*, 73(2):213–238, 2007.

[38] X. Zhen, L. Shao, D. Tao, and X. Li. Embedding motion and structure features for action recognition. *IEEE TCSVT*, 23(7):1182–1190, 2013.