Human Action Recognition by Random Features and Hand-Crafted Features: A Comparative Study

Haocheng Shen[1], Jianguo Zhang[1], and Hui Zhang[2]

[1] School of Computing, University of Dundee, Dundee, United Kingdom
hyshen@dundee.ac.uk
jgzhang@computing.dundee.ac.uk
[2] Department of Computer Science & Technology, United International College, Zhuhai, China
amyzhang@uic.edu.hk

Abstract. The popular approach for human action recognition is to extract features from videos as representation, followed by a classification of these video representations. In this paper, we investigate and compare hand-crafted and random feature representation for human action recognition on YouTube dataset. The former is built on 3D HoG/HoF and SIFT descriptors while the latter bases on random projection. Three encoding methods: Bag of Feature(BoF), Sparse Coding(SC) and VLAD are adopted. Spatial temporal pyramid and a two-layer SVM classifier are employed for classification. Our results demonstrate that: 1) Sparse Coding is confirmed to outperform Bag of Feature; 2) Using a model of hybrid features incorporating frame-static can significantly improve the overall recognition accuracy; 3)The frame-static features works surprisingly better than motion features only; 4) Compared with the success of hand-crafted feature representation, the random feature representation does not perform well in this dataset.

Keywords: Action Recognition, Hand-crafted Feature, Random Representation

1 Introduction

Recognizing human action is a significant branch of computer vision and attracting increasing attentions due to its widely applications like crime monitoring and human-computer interaction. Generally, the recognition task can be simply viewed as a combination of two subtasks: extract features as representations from video frame sequence, and subsequent classification of the video representations. Among the two subtasks, one key point is to built such a feature representations, which contain the main structure of an action and robust to background cluttering, illumination and scale changes etc. Substantial approaches of exploring the feature representation have been proposed and proven successful, such as 3D HoG [8], HoG/HoF [9], extended SURF [17]. These feature representations are

all hand-crafted and need to be computed by a specific mathematical manner. Recently, a method based on random feature representation has been popular in texture recognition [14], face recognition [18], and medical image analysis [12]. However, little work has been reported on applying this representation into video based action recognition. Therefore, in this paper, we evaluate and compare these two different feature representations for action recognition task. We have three main contributions: (1) a comparative study of different combinations of existing schemes for video action recognition based on hand-crafted feature representation and report the best combination whose performance is competitive to one of the state-of-art techniques on the same dataset; (2) Investigate the popular random feature representation to see whether it is a feasible approach for video based human action recognition; (3) Investigate the role of frame-static features and motion features for action recognition on the popular YouTube dataset.

The rest of this paper is organized as follow: Section 2 reviews relevant literature of approaches for action recognition; Section 3 describes each component of designed algorithm in details; Section 4 indicates the implementations and experiment results; Conclusions and future work are given in Section 5.

2 Related Work

The approach for action representations can be generally divided into two categories: global representations and local representations. For the former, the human body is first located in the image. Then the person referred as interest of region (ROI) would be encode as a whole, resulting in the image descriptors. The common approach is silhouettes [2] and optic flow [7]. Local representation is a more popular approach which describes the observation as a collection of local descriptors or patches. The cuboid descriptor [4], HoG/HoF descriptor [9], 3D HoG descriptor [8] and extended SURF [17] are all robust in action recognition.

3 Method

Considering the large variation in realistic videos, static feature like a static pose in a single image also contains important action contextual information, such as boating in the water, riding in the street or on a grass land. Those can provide strong cues and thus serve as a complementary of motion feature for action recognition. Motivated by this observation, in this paper, we investigate both feature representations, and build a hybrid model upon them. The flowchart of our video based action recognition is shown in Figure 1. We will follow this flowchart to describe our algorithm in details step by step.

3.1 Spatial-Temporal Interest Points Detection

Spatial-temporal interest points are the locations in space and time domain where a significant variation occurs in the local neighborhood. We apply the

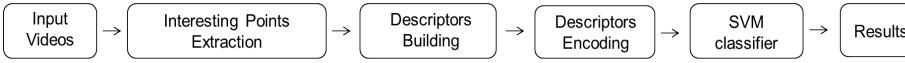


Fig. 1. The flowchart of video based recognition.

extension of Gabor filter proposed by Dollar et al. [4] to extract the 3D interest cuboids, which captures the most important characteristics of the movement occurring in the video. The response function has the form:

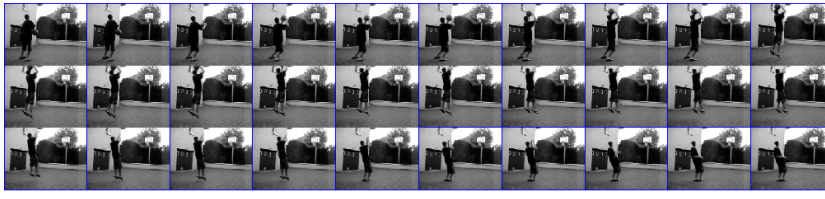
$$R = (I * g * h_{ev})^2 + (I * g * h_{od})^2 \tag{1}$$

where $g(x, y; \sigma)$ is the 2D Gaussian smoothing kernel for spatial dimensions, and h_{ev} and h_{od} are a quadrature pair of 1D Gabor filters applied temporally. They are defined as:

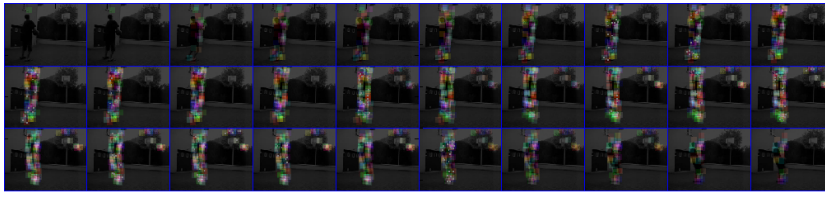
$$h_{ev}(t; \tau, \omega) = -cos(2\pi t\omega)e^{-t^2/\tau^2} \tag{2}$$

$$h_{od}(t; \tau, \omega) = -sin(2\pi t\omega)e^{-t^2/\tau^2} \tag{3}$$

where $\omega = 4/\tau$. The interest points are located in the local maxima corresponding to the response function. The parameter σ and τ correspond to the spatial and temporal scale of the detected cuboid. We set the size of the cuboid to 19 $\times$ 19 $\times$ 11 pixels. Some examples of interest cuboids detected by the 3D Gabor detector on video frame sequence are shown in Figure 2.



(a)



(b)

Fig. 2. An example 3D interest cuboids detection. (a) original frames; (b) cuboids detected by 3D Gabor filter. Best viewed in colour

3.2 Hand-crafted Feature Representation

The visual content of a video segment can be represented by a set of descriptors computed at every interest point position within its near cuboid region. It is obvious that the oriented gradient can capture spatial information while optic flow is able to catch the movement information. Therefore, we adopt the 3D HoG/HoF descriptor similar to Laptev et al. [9], which computes histograms of both oriented gradient and optic flow accumulated in spatial-temporal interest cuboids.

Specifically, the 3D interest cuboid is firstly smoothed and divided into $3 \times 3 \times 2$ grid of cells; for each cell, 4-bin histograms of gradient (HoG) and 5-bin histograms of optic flow (HoF) are calculated based on the oriented direction. Then the normalized histograms from each small grid are concatenated to form the local descriptor. We employ PCA to reduce the dimensionality to 200 experimentally.

Using *motion feature* only may not be distinct enough, especially for the unrestricted videos like YouTube action dataset. Intuitively, The static feature can be viewed as a very strong complementary. To extract *static feature*, we sample temporally at every 15 frames from the frame sequence of the video. For each frame, dense sampling is applied to extract the interest points upon which SIFT descriptors are built. Additionally, multi-scale static feature is achieved by changing the size of the static image by multiplying $1/\sqrt{2}$.

3.3 Random Feature Representation

We employ *random projection* to build random feature representation. The key idea of random projection originated from the Johnson-Lindenstrauss lemma: if points in a high dimension are projected onto a randomly selected subspace of suitable dimension, then the distance between points are approximately preserved. In practice, the original d-dimensional data is projected to a k-dimensional ($k << d$) subspace using a random matrix $k \times d$ matrix $\mathbf{R}$ whose columns have unit lengths. It can be represented by:

$$\mathbf{X}^{RP}_{k\times N} = \mathbf{R}_{k\times d}\mathbf{X}_{d\times N} \tag{4}$$

As before, we apply the random projection on both motion feature and static feature to form the random feature representation. Specifically, for each extracted cuboid, we first normalize the intensity of each pixel within the cuboid and then uniformly divide the cuboid into $2 \times 2 \times 2$ grids. Assume the size of each grid is $w \times h \times t$ pixels so for each grid we identify gray-scale vector $\mathbf{v} \in \mathbb{R}^d (d = wht)$ by stacking its columns; then random projection is applied on this gray-scale vector to reduce dimensionality and form the random feature descriptor. The random matrix $\mathbf{R}$ is defined as the Gaussian measurement matrix whose elements are independent, zero-mean, unit-variance Gaussian random variables. Finally the projected vectors for each sub-cuboid are concatenated to form the local descriptor of the whole cuboid.

Similarly for the static feature extraction, we use dense sampling as before on each sampled video frame sequence. For each dense point, the patch whose size is the same as that of SIFT descriptors is extracted and the gray-scale vector is formed by stacking its columns. Then random projection is employed to generate the local static random descriptors.

3.4 Descriptors Encoding

As the number of local descriptors extracted by the above methods varies from each video, distinguishing these descriptors from different classes of action directly is not a straightforward task. A popular approach is to firstly learn a codebook containing a fixed number of visual words based on the training descriptors set, then encode the descriptors with the codebook.

A simple but effective method to learn the codebook is K-means clustering algorithm. The main idea is to minimize the sum of squared Euclidean distances between points $\mathbf{x}_j$ and their nearest cluster $\mathbf{v}_k$:

$$\arg\min_{\mathbf{V}} \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in \mathbf{V}_i} \|\mathbf{x}_j - \mathbf{v}_k\|^2 \tag{5}$$

where $\mathbf{V} = [\mathbf{v}_1, ..., \mathbf{v}_k]^\top$ are the target codebook with K cluster centers. We propose 2-level K-means clustering to generate the codebook: for each class of action, apply K-means for the first level clustering, then based on the first level results, the K-means clustering is applied again to create the final codebook. The size of codebook is set to 256.

We mainly evaluate two popular encoding methods: **Bag of Feature** and **Sparse Coding** for both feature representations. Moreover, we extra evaluate **Vector of Locally Aggregated Descriptors** (VLAD) for random feature representation.

Bag of Feature Let $\mathbf{X}$ be a set of descriptors in a D-dimensional feature space, $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_M]^\top \in \mathbb{R}^{M \times D}$. The Bag of Feature quantization problem can be re-formulated into a matrix factorization problem:

$$\begin{aligned} \min_{\mathbf{U}} \quad & \sum_{m=1}^{M} \|\mathbf{x}_m - \mathbf{u}_m \mathbf{V}\|^2 \\ \text{subject to} \quad & Card(\mathbf{u_m}) = 1, |\mathbf{u_m}| = 1, \mathbf{u}_m \geqslant 0, \forall m \end{aligned} \tag{6}$$

where $\mathbf{U} = [\mathbf{u}_1, ..., \mathbf{u}_M]^\top$ is the cluster membership indicators and $\mathbf{V}$ is the pre-calculated codebook. The cardinality constraint $Card(\mathbf{u_m}) = 1$ means that only one element of $\mathbf{u}_m$ is nonzero, and $|\mathbf{u_m}|$ indicates that the summation of the absolute value of each element in $\mathbf{u}_m$. After obtaining the encoded descriptor set $\mathbf{U}$, the video can be represented by frequencies of each visual word. Since the number of visual words is fixed for all descriptors sets, a video with arbitrary

number of descriptors is then converted into a single histogram vector whose length equals to the number of visual words. This provides extreme convenience for the future classification processing.

Sparse Coding The constraint for BoF model $Card(\mathbf{u_m}) = 1$ is too restrictive to reconstruction $\mathbf{X}$ with low error. We can relax the constraint by making $\mathbf{u_m}$ to have a small number of nonzero element. Meanwhile, the number of nonzero element is enforced to be minimum. Then the BoF is turned into another problem known as Sparse Coding:

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V}} \quad & \sum_{m=1}^{M} \|\mathbf{x}_m - \mathbf{u}_m\mathbf{V}\|^2 + \lambda\,|\mathbf{u_m}| \\ \text{subject to} \quad & \|\mathbf{v_k}\| \leqslant 1, \forall k = 1, 2, ..., K \end{aligned} \tag{7}$$

Similar to BoF, in the training stage a set of training descriptors are used to solve Equation 7 with respect to $\mathbf{U}$ and $\mathbf{V}$. The conventional way for such a optimization problem is to iteratively optimize either over $\mathbf{U}$ or $\mathbf{V}$ while fixing the other. We set the initial codebook $\mathbf{V}$ of Sparse Coding as the result generated by K-means algorithm described above instead of using a random initialization. This processing can make the objective function more optimized when the number of iteration is fixed.

Each column of $\mathbf{U}$ in Sparse Coding corresponds to the coefficients of all the local descriptors to one specific visual word in the codebook $\mathbf{V}$, we adopt the max pooling function for SC, which has been well established by biophysical evidence and empirically justified by many image categorization algorithms. It can be represented by:

$$z_j = \max\left\{|u_{1,j}|, |u_{2,j}|, ..., |u_{M,j}|\right\} \tag{8}$$

where z_j is the j-th element of $\mathbf{z}$, $u_{i,j}$ is the matrix element at i-th row and j-th column of $\mathbf{U}$.

Vector of Locally Aggregated Descriptors Besides BoF and SC, we evaluate another encoding method: vector of locally aggregated descriptors (VLAD) [6] for random feature representation. Similar to BoF, we first learn a codebook $\mathbf{V} = [\mathbf{v}_1, ..., \mathbf{v}_k]^\top$ by K-means clustering algorithm. The idea of the VLAD is to accumulate the difference between each visual $\mathbf{v}_i$ and the descriptor $\mathbf{x}_i$ which is assigned to that visual word. Therefore, if the local descriptor is d-dimensional, the dimension D of VLAD would be $D = k \times d$. A component $u_{i,j}$ of VLAD can be obtained by summing over all the local random feature descriptors:

$$u_{i,j} = \sum_{\mathbf{x} \text{ belong to } \mathbf{v_i}} x_j - v_{i,j} \tag{9}$$

where the indices $i = 1...k$ and $j = 1...d$ index the visual word and the local descriptor component respectively.

3.5 Spatial-Temporal Pyramid

All the encoding methods described above only capture the statistical characteristic of the descriptors set. None of spatial and temporal layout of geometrical features has been taken into consideration. Spatial Pyramid Matching (SPM) proposed by [10] overcomes this limitation in still image classification. It works by partitioning the image into increasingly fine sub-regions and computes histograms of local descriptors over the resulting sub-regions. The final feature vector is formed by concatenating histograms of each sub-region with the corresponding weight of each level of pyramid.

The spatial pyramid is a simple and computationally efficient complement of an orderless BoF image representation. It has shown significantly improved performance over the standard BoF model as this method describes the observations as a collection of local representations, which are somewhat invariant to changes in scale, illumination and partial occlusions. We extend this approach to 3D by adding subregions with respect to temporal domain. The spatial-temporal pyramid is built by uniformly dividing the frame sequence of video into $2 \times 2 \times 2$ grids for the first level and $3 \times 3 \times 3$ grids for the second level. The descriptor set of each subregion is a set of descriptors whose corresponding interest points are located within such a subregion. Then the local characteristics in totally 36 subregions are calculated by BoF or SC or VLAD with corresponding local descriptors set. Finally, for BoF or VLAD, STP concatenates histograms or matrix of each subregion of video multiplied by the weights corresponding to pyramid level to form a feature vector of the video, while for Sparse Coding, the corresponding coefficients to the local descriptor sets in each subregion are computed and concatenated, then the max pooling function is applied to form such a vector of the video.

3.6 Support Vector Machine

The size of feature vectors of videos generated by Spatial-Temporal Pyramid (STP) approach would be very large. For example, a feature vector of a video constructed by 3-level uniformly distributed pyramid and 256 visual words would have 9216 attributes. If these feature vectors are directly classified by SVM classifiers, it would be very computationally expensive on both training and testing stage, especially for large dataset used which involves more than a thousand videos.

We build a two-layer SVM classifiers system for classification processing. The structure of the two-layer SVM classifiers system is shown in Figure 3. In the first layer, the vectors produced by the same pyramid level in different videos are classified separately using non-linear SVM classifiers. The decision values outputted by the first layer for each video against the corresponding class label can be viewed as an abstract descriptors of the particular pyramid level of videos. Then the decision values from each pyramid level are concatenated and classified again by another non-linear SVM classifier.

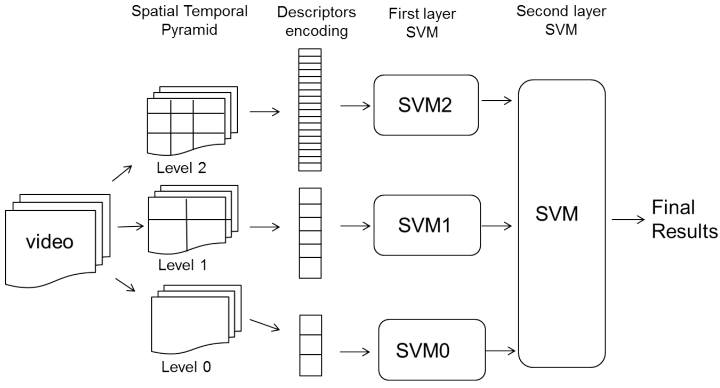


Fig. 3. The two layer SVM classifier structure

The two-layer SVM classifiers have the following attractive properties: 1) The decision values represent the descriptors set in a more concise way and are more robust to the effect of noise. 2) The number of decision values only relate to the number of action classes. For example, M action class would generate $M(M-1)/2$ binary classifiers in each pyramid level for pairwise classification. It can significantly reduce the dimensional of the descriptors as well as the computation time. 3) The process of classification on each pyramid level are independent which enables the parallelized computing for the whole process. 4) the second layer SVM assigns weights based on action classes for each pyramid level instead of assigning it to the visual words of different levels directly. This leads to better results than the standard SPM method with traditional one-layer SVM classifier.

4 Experiments

4.1 Dataset

The video dataset we used is the YouTube action dataset from [13]. The videos in this dataset are mostly collected from YouTube and captured under uncontrolled condition so they contain significant camera motion, background clutter, illumination changes, viewpoint changes and objects scale changes. All these properties of this video dataset make it closer to the realistic video data in our daily life, but also push correct recognition more highly challenging.

YouTube action dataset contains 11 action categories: basketball *shooting*, cycling, diving, golf *swing*, horse-back *riding*, soccer *juggling*, swinging, tennis *swinging*, trampoline *jumping*, volleyball *spiking* and *walking* with a dog. In order to remove the unfair effect of the same background in recognition, the videos in each action are organized into 25 relatively independent groups, where each group is taken in different actors, backgrounds, viewpoints. Our experiments setup is the same as that proposed in [13]. There are totally 1168 videos for

use. In the training phase, leave-one-out group cross validation is used. All the colorful videos are convert into gray-level in advance before further processing.

4.2 Hand-crafted Feature Representation

Firstly, we evaluated BoF and SC encoding combined with spatial temporal pyramid based on the motion feature only. The results are shown in Figure 4. As expected, it can be observed that SC achieves higher accuracies in most classes of action as well as the overall accuracy 64.98% than that 60.10% of BoF. We explained this improvement as that SC can achieve a much lower reconstruction error due to the less restrictive constraint, although it is more computationally expensive.

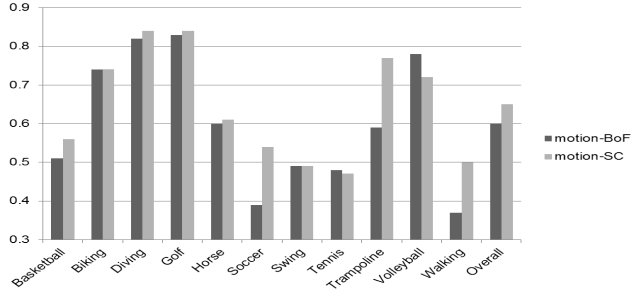


Fig. 4. The classification accuracies generated by BoF and SC based on the motion feature only.

The number of static local descriptors can be tens of thousands. Because of the high memory requirement, the static feature is built by only BoF due to its low computational complexity. The overall accuracy based on static feature built on original frames is 65.33%, while the accuracy based on static feature built on multi-scale frames is 66.52%. There is no significant improvement between multi-scale and original static feature. Therefore, we discard multi-scale static feature for reducing the computational complexity and use the original static feature only for the rest experiments.

We also evaluated the motion model, static model and hybrid model. The motion model is only based on motion feature encoded by SC while the static model is only based on static feature encoded by BoF. The hybrid model is to combine the motion feature and static feature. The results are shown in Figure 5. Intuitively, motion feature and static feature are complementary for action recognition. And this has been proven by our experiment that the accuracy of hybrid model is higher than both motion and static model in every class of action recognition as well as the overall accuracy, which is 75.51%, 64.98%, 65.33% for hybrid, motion and static model respectively. The hybrid model has the better performance over 10% than both motion and static model, which is impressive.

Hence, it can be concluded that the hybrid model can achieve the best results, and not only motion feature but also static feature plays a significant role in action recognition. It can be also observed that the static features works surprisingly better than motion features only. We explain this improvement by the fact that the dense feature outperforms the feature based on interest points. The confusion table for classification using our proposed hybrid model can be seen in Figure 6.

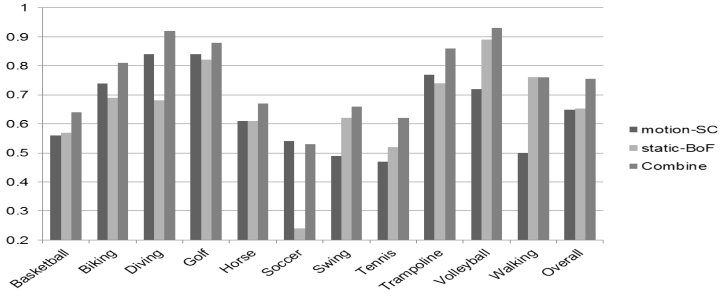


Fig. 5. The classification accuracies generated by motion, static and hybrid model.

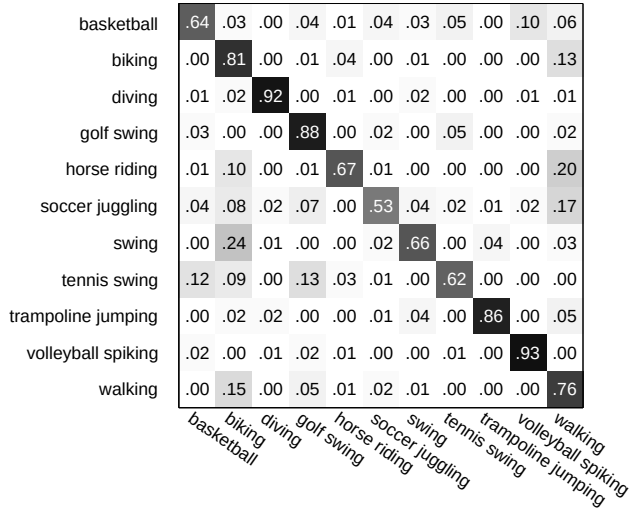


Fig. 6. The confusion table for classification using hybrid model.

Lastly, we compared our method based on hand-crafted feature representation with the state-of-art on the same dataset (see Table 1). It can be clearly seen that our method is competitive with the state-of-art. Specifically, our framework is quite similar with Liu et al. [13] but our overall accuracy (75.51%) is higher than theirs (71.2%). Note that the highest accuracy (85.4%) proposed by Wang et al. [16] is much higher (over 10%) than all other methods because they adopted the dense trajectories feature on building motion feature, which is very computational intensive and memory consuming.

Liu et al. (2009) [13]	**71.2%**
Ikizler-Cinbis and Sclaroff (2010) [5]	75.21%
Brendel and Todorovic (2010) [3]	77.8%
Le et al. (2011) [11]	75.8%
Bhattacharya et al. (2011) [1]	76.5%
Wang et al. (2013) [16]	85.4%
Our method	**75.51%**

Table 1. The comparison between our method and the state-of-art.

4.3 Random Feature Representation

The parameter settings for building random feature representation is almost the same as building the hand-crafted feature representation described above. We also evaluated the algorithms on a range of parameter values to search appropriate parameters to achieve the highest possible accuracies. Firstly, we searched for the appropriate projected dimension n. The sub-feature vector is projected into 25, 50, 100, 200 dimensions so that the dimensionality of the final local descriptor would be 200, 400, 800 and 1600 respectively. Note that for the descriptors with 1600 dimensionality, we sampled 400 descriptors from each video to generate the codebook due to the high memory requirement. Another parameter we try to optimize is the size of the cuboid, the size employed in building hand-crafted feature representation ($19 \times 19 \times 11$ pixels) is taken as the benchmark.

The results based on the descriptors projected to diverse dimensions by random projection on the extracted cuboid are shown in Figure 7. It can be seen that the accuracies over diverse dimensions of descriptors are all fluctuated around 50% and there is no significant difference between each dimensionality. In addition, no obvious tendency of improvement or decreasing over the diverse dimensions can be observed. Therefore, we conclude that the projected dimension is not an important fact that affects the final classification accuracy. The projected dimension is then fixed to 200 as same as hand-crafted descriptors due to its lower computational complexity and for the sake of comparisons.

To investigate the effect of size of cuboid, we conducted a set of experiments based on 3 different spatial size and 2 different temporal size. The results of total

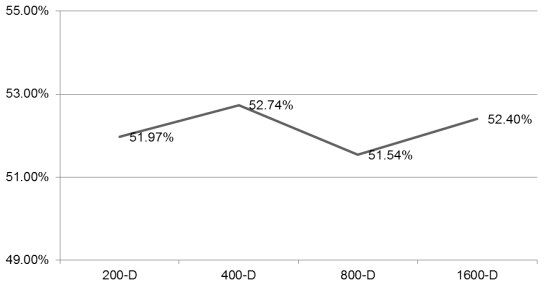


Fig. 7. The accuracies based on the descriptors projected to 200, 400, 800 and 1600 dimensions by random projection on the extracted cuboid with the size of $19 \times 19 \times 11$ pixels.

6 experiments with different cuboid sizes are shown in Table 2. Again, all the results fluctuated around 50% and there is no significant improvement among them. The best result we got is 51.97% with the $19 \times 19 \times 11$ pixels cuboid size. Therefore, changing the size of the extracted cuboid would not improve the performance.

In addition, the result applying VLAD encoding method is 55.65% based on the 200 dimension random descriptors and 128 visual words, which is similar to that of using SC (55.31%). As expected, the result of VLAD is better than BoF (51.97%) and the computational time is much less than Sparse Coding but at the cost of consuming memory.

$11 \times 11 \times 11$	$11 \times 11 \times 23$
50.26%	45.89%
$19 \times 19 \times 11$	$19 \times 19 \times 23$
51.97%	48.03%
$39 \times 39 \times 11$	$39 \times 39 \times 23$
44.09%	46.40%

Table 2. The results based on different sizes of the cuboid with a fixed projected 200 dimension.

4.4 Comparisons

We also evaluated the random descriptors on the static model and the hybrid model. Again, the hybrid model can achieve about 8% improvement on overall accuracy over the motion and static model. The best results we obtained for random feature representation and the corresponding results generated by the hand-crafted feature are list in Table 3.

Method	Random Feature	Hand-crafted Feature
motion - BoF	51.97%	**60.10%**
motion - SC	55.31%	**64.98%**
static - BoF	51.54%	**65.33%**
Combined	62.50%	**75.51%**

Table 3. The comparison between the random feature and the hand-crafted feature representation.

From the Table 3, we can see that there is a big difference between results from the two proposed feature representations. For each evaluation of encoding method, the performance of hand-crafted feature is over 10% higher than that of random feature. As the framework, encoding and classifiers parameter settings are totally the same for evaluating both feature representation, we can conclude that the random feature representation does not perform well in this YouTube action dataset although it is simple to be implemented and successful in other recognition domains like texture recognition. Recall that random projection is a power tool in dimensionality reduction and should be beneficial in the cases where the distances of the original high dimensional data are meaningful. Therefore, we explained this failure of random feature representation for possibly one reason that the original distance or similarities information contained by the extracted cuboids are themselves suspect so that the random feature descriptors are not distinct enough to be classified.

5 Conclusions

In this paper, we investigate and compare two different feature representations for video based human action recognition: hand-crafted and random feature representation. The former is built by 3D HoG/HoF descriptors for motion feature and SIFT descriptors for static feature while the latter is based on random projection. Three popular approaches of encoding descriptors: BoF, SC and VLAD are all applied in our experiments. Additionally, spatial temporal pyramid and a two layer SVM classifier are employed for classification processing.

For the motion feature of both representations, we evaluated both BoF and SC encoding methods. The results confirms that SC outperforms BoF as indicated in object recognition community. Based on the performance of the motion, static and hybrid model, we found that using hybrid features of motion and static can significantly improve the overall recognition accuracy which only uses motion features. It can be concluded that as complementary of the motion feature, the static feature plays an essential role in action recognition on this dataset, which should be paid more attention to in future relevant researches. Compared with the success of the popular hand-crafted feature representation such as 3D HoG/HoF, SIFT descriptors for action recognition, the proposed random feature representation based on random projection does not perform well in this

dataset. This is probably due to the suspect of original information contained by the extracted cuboids as well as the random error.

The overall accuracies over YouTube action dataset based on random features is far behind the state-of-art performance. For the future work, the random feature based approach would be experimented on other datasets, such as KTH action dataset [15] and Hollywood movie dataset [9].

References

1. Bhattacharya, S., Sukthankar, R., Jin, R., Shah, M.: A probabilistic representation for efficient large scale visual recognition tasks. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. pp. 2593–2600. IEEE (2011)
2. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on. vol. 2, pp. 1395–1402. IEEE (2005)
3. Brendel, W., Todorovic, S.: Activities as time series of human postures (2010)
4. Dollár, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on. pp. 65–72. IEEE (2005)
5. Ikizler-Cinbis, N., Sclaroff, S.: Object, scene and actions: Combining multiple features for human action recognition. In: Computer Vision–ECCV 2010, pp. 494–507. Springer (2010)
6. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. pp. 3304–3311. IEEE (2010)
7. Ke, Y., Sukthankar, R., Hebert, M.: Spatio-temporal shape and flow correlation for action recognition. In: The VS Workshop, Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on. pp. 1–8. IEEE (2007)
8. Klaser, A., Marszalek, M.: A spatio-temporal descriptor based on 3d-gradients. In: British Machine Vision Conference, 2009. BMVC 2009. IEEE Conference on
9. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. pp. 1–8. IEEE (2008)
10. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. vol. 2, pp. 2169–2178. IEEE (2006)
11. Le, Q.V., Zou, W.Y., Yeung, S.Y., Ng, A.Y.: Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. pp. 3361–3368. IEEE (2011)
12. Li, W., Zhang, J., McKenna, S.J., Coats, M., Carey, F.A.: Classification of colorectal polyp regions in optical projection tomography. ISBI (2013)
13. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos in the wild. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 1996–2003. IEEE (2009)
14. Liu, L., Fieguth, P.: Texture classification from random features. Pattern Analysis and Machine Intelligence, IEEE Transactions on 34(3), 574–586 (2012)

15. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local svm approach. In: Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on. vol. 3, pp. 32–36. IEEE (2004)
16. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Dense trajectories and motion boundary descriptors for action recognition. International Journal of Computer Vision pp. 1–20 (2013)
17. Willems, G., Tuytelaars, T., Van Gool, L.: An efficient dense and scale-invariant spatio-temporal interest point detector. In: Computer Vision–ECCV 2008, pp. 650–663. Springer (2008)
18. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. Pattern Analysis and Machine Intelligence, IEEE Transactions on 31(2), 210–227 (2009)
19. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 1794–1801. IEEE (2009)