International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:6, No:11, 2012

Evaluation of Classifiers Based on I2C Distance for Action Recognition

Lei Zhang, Tao Wang, and Xiantong Zhen

***Abstract*—**Naive Bayes Nearest Neighbor (NBNN) and its variants, i,e., local NBNN and the NBNN kernels, are local feature-based classifiers that have achieved impressive performance in image classification. By exploiting instance-to-class (I2C) distances (instance means image/video in image/video classification), they avoid quantization errors of local image descriptors in the bag of words (BoW) model. However, the performances of NBNN, local NBNN and the NBNN kernels have not been validated on video analysis. In this paper, we introduce these three classifiers into human action recognition and conduct comprehensive experiments on the benchmark KTH and the realistic HMDB datasets. The results shows that those I2C based classifiers consistently outperform the SVM classifier with the BoW model.

***Keywords*—**Instance-to-class distance, NBNN, Local NBNN, NBNN kernel.

I. Introduction

RECENTLY, sparse representations based on detected spatiotemporal interest points are drawing much attention. Human action recognition systems based on the bag of words (BoW) model have achieved good results in many tasks. The BoW model has many advantages, such as being less sensitive to partial occlusions and clutter and avoiding some preliminary steps, e.g. background subtraction and target tracking in holistic methods. Nevertheless, this model also has deficiencies. A key limitation of them lies in its inability to capture adequate spatial and temporal information. Since the BoW model is actually based on mapping local features of each video sequence onto a pre-learned dictionary, it inevitably introduces quantization errors during the codebook creation and the errors would be propagated to the final representation and degrade the recognition performance. Moreover, codewords, as the cluster centers, obtained by k-means clustering, would gather around dense regions of the local feature space, which, unfortunately, makes the codewords less effective as the action primitives. Additionally, the size of the dictionary needs to be empirically determined, which is less flexible for different tasks.

Instead, by exploring the image-to-class distance, Boiman proposed a NBNN (Naive Bayes Nearest Neighbor) classifier [1], which avoids the quantization errors in the BoW model and obtains impressive performance in image classification. Based on NBNN classifier, Tuytelaars combined the idea of image-to-class distance with o SVM and proposed the NBNN kernels [2], which considered both local and global effects in image processing.

Additionally, locality is a hot topic in image and video processing recently. Gao proposed Laplacian sparse coding [3] for image classification, in which the similarity among the sparse codes is considered during the process of sparse coding. Wang proposed locality-constrained linear coding (LLC) [4] for image classification, which achieves better performance than sparse coding by projecting the descriptor into its local-coordinate system. Liu [5] proposed a 'localized' soft assignment to improve the performance to the level of sparse coding or even better than sparse coding. Similarly, locality constraint is introduced into NBNN by McCann [6], which is local NBNN approach.

Most of above works are conducted in image processing. To our knowledge, related works are seldom applied on video analysis. In video processing, we rename the distance as instance-to-class (I2C) which can cover both image and video processing. In this paper, we evaluate those I2C based classifiers performances on action recognition, i.e., NBNN, the local NBNN and the NBNN kernels. We conduct the experiments on both KTH and HMDB. KTH [7] is the most popular benchmark dataset and HMDB [8] is the newly released dataset with realistic actions.

II. Revisit About Distance

In video processing, there are two ways to deal with video information. One is to treat the video as the frame sequence, and the descriptor is then based on 2-D plane (frame), as SIFT [9]. The other one is to view the video as a 3-D signal, and the descriptor is based on 3-D cuboid, as hog/hof on harris 3-D detector [10]. No matter which kind of approach, the descriptor can represent not only the interested point, but also the neighborhood information around this point patch information). The local processing approaches represent the video content just by the set of descriptors, and for each video, the number of descriptor may be different.

In order to eliminate the effect of different number of descriptors in each video and obtain the information from a higher level (distribution), histogram is built for each video.

No matter the distance is based on descriptor or histogram, distance can be divided into following classes by its basic element. [11] also gives some analysis about distance definition in image processing.

Patch-to-patch distances: pixel is the smallest unit in image and video processing. Since the less quantity information in pixel, normally, descriptor is extracted based on a patch/cuboid of interested point. Patch-to-patch distance can be described as

Lei Zhang and Tao Wang are from information and communication engineering college, Harbin Engineering University, China (e-mail: zhanglei@hrbeu.edu.cn).

XianTong Zhen is Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield, UK (e-mail: zhenxt@gmail.com).

$dis(d_i, d_j)$, where d_* is a descriptor in high dimension space.

Patch-to-instance distance: instance here means image in image processing and video in video processing. Since instance contains more than one descriptor, it can be viewed as the sum or the minimum of patch-to-patch distance as (1).

$$dis(d_i, I) = \min_{j \in I} dis(d_i, d_j) \tag{1}$$

Instance-to-instance distance: For each descriptor in instance I_1, the sum of each patch-to-patch distance is viewed as the instance-to-instance distance, which is as (2).

$$dis(I_1, I_2) = \sum_{i \in I_1} \min_{j \in I_2} dis(d_i, d_j) = \sum_{i \in I_1} dis(d_i, I) \tag{2}$$

Instance-to-class distance:

$$dis(I_1, C) = \sum_{i \in I_1} dis(d_i, d_i^c) = \sum_{i \in I_1} \min_{j \in c} dis(d_i, d_j) \tag{3}$$

Where d_i^c means the nearest neighbor descriptor of d_i in class c.

Most classifiers, like SVM in action recognition, are based on instance-to-instance distance. For some applications, such as videos in HMDB, divergence of instance-to-instance distance may be huge, which can be drawn from Fig. 1. There are two samples for cartwheel actions, and the upper one happens indoor while the lower one is out-door with different background. For the target person's size, the lower one is smaller and the action's direction is different from the upper one. Even for the same action, the appearances of these actions are obviously different. If we limit to learn one model across all instances, it could reduce our ability to determine similarity of test sample to these ones.

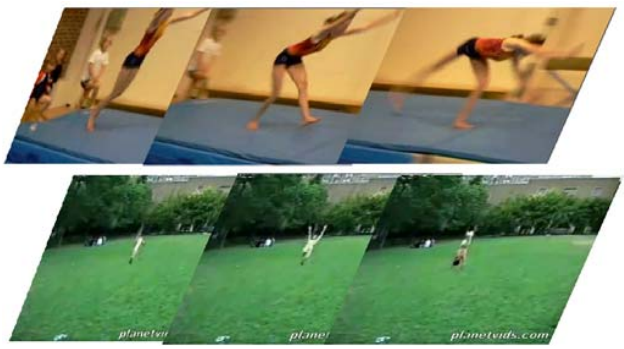

Fig. 1 Two samples of cartwheel action in HMDB corpus, the upper one is indoor with bigger size of target person, while the lower one happens outdoor with small size of target person

While for NBNN, its most advantage lies in its instance-to-class distance. This kind of distance tolerates dissimilarities between instances in each class to some extent, as shown in Fig. 2. With regard of instance-to-instance distance, for every feature (descriptor) in instance *i*, it can only find the nearest neighbor from another instance, like *j* in Fig. 2. That is why for yellow square (descriptor in instance *i*) even if there are some blue circles (other descriptors in the same class) closer than blue square (descriptor in instance *j*), the distance is still shown as the black line. But for instance-to-class distance, it breaks this limitation. For finding the nearest neighbor of one descriptor, the range can be enlarged to all the descriptors in the same class to compute this descriptor-to-class distance, like the red line in Fig. 2. Accumulating all the descriptor-to-class distances of one instance, it forms the instance-to-class distance.

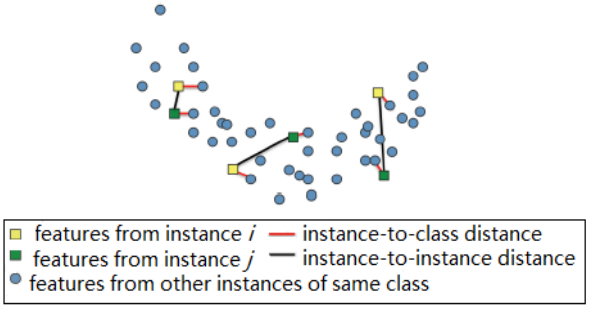

Fig. 2 The instance-to-class concept: even though the features of the two instance are not very similar (close), their distances to the class distribution are similar, and that is what counts for NBNN [1]

III. Classifier on Instance-to-Class Distance

We investigate the effects of recent proposed classifiers based on instance-to-class distance on action recognition, including NBNN, Local NBNN and NBNN kernel combined with SVM.

A. Notation

$C = \{c_1, ..., c_n\}$ is the class set, and $D = \{d_1, ..., d_m\}$ is the descriptor set. $p(d \mid c_i)$ is the distribution of descriptor in class *i*. Similarly, $p(d \mid q)$ is the distribution of descriptor in test video q.

B. NBNN

Given a query instance q, its class label can be obtained by minimizing KL (Kullback-Leibler) distance as (4).

$$\hat{c} = \arg\min_c KL(p(d \mid q) \,\|\, p(d \mid c)) \tag{4}$$

Where KL distance is as (5).

$$KL(p(d \mid q) \,\|\, p(d \mid c)) = \sum_i p(d_i \mid q) \log \frac{p(d_i \mid q)}{p(d_i \mid c)} \tag{5}$$

In [4], (4) is proved to satisfy maximum-a-posteriori (MAP) criterion under the Naive-Bayes assumption. Considering the class prior probability is uniform, then MAP estimation changes into maximum likelihood (ML) estimation.

Neglecting the terms with no relationship with class label in (4) and (5), (4) can be turned into:

$$\hat{c} = \arg\max_c \sum_i p(d_i \mid q) \log p(d_i \mid c) \tag{6}$$

For the distribution in (6), a Parzen density estimation provides an accurate non-parametric approximation of the continuous descriptor probability density as (7).

$$p(d \mid c) = \frac{1}{L}\sum_{j=1}^{L} K(d - d_j^{\,c}) \tag{7}$$

Where

$$K(d - d_j^{\,c}) = \exp(-\frac{1}{2\sigma^2} \| d - d_j^{\,c} \|^2) \tag{8}$$

(7) covers all descriptors ($d_j^{\,c}$) in class c and L is the total number of these descriptors. In theory, when L closes to infinite, the accurate of the distribution can be estimated. In NBNN, there is another assumption of using r-nearest neighbors to replace the whole summation in (7). Furthermore, if r equals to unit, $d_j^{\,c}$ is represented as $d_{NN}^{\,c}$, which is the nearest-neighbor of d in class c, then (6) can be turned into:

$$\hat{c} = \arg\max_{c_i} \sum_i p(d_i \mid q)(-\frac{1}{2\sigma^2} \| d_i - d_{NN}^{\,c} \|^2) \tag{9}$$

Furthermore, suppose $p(d_i \mid q)$ keeping unchanged for different i and the same kernel bandwidth (σ) with different class c, then this equation can be simplified as :

$$\hat{c} = \arg\min_{c_i} \sum_i (\| d_i - d_{NN}^{\,c} \|^2) \tag{10}$$

This equation in fact is similar with instance-to-class distance as (3).

C. Local NBNN

In NBNN, we compute all the distances of descriptors in test video to all classes, and find the class with minimum one. In local NBNN, the idea is a little different from NBNN. Not all classes will attend in the computation for d_i in (4), but the classes with the k-nearest-neighbor of d_i.

This search strategy can speed up the algorithm, and also achieve better classification performance by ignoring the distances to classes far from the test descriptor. The algorithm is described as follow:

Algorithm LOCAL NBNN

for all descriptor $d_i \in q$ **do**

find the k+1 nearest neighbor in all descriptor in all classes $\{p_1, p_2, ..., p_{k+1}\}$

$dist_B \leftarrow \| d_i - p_{k+1} \|^2$

for all categories C found in k+1 nearest neighbor **do**

$dist_C \leftarrow \min_{\{p_j | class(p_j)=c\}} \| d_i - p_j \|^2$

$total[c] \leftarrow total[c] + dist_C - dist_B$

end for

end for

$\hat{c} = \arg\min_c total[c]$

return $\hat{c}$

D. The NBNN Kernels

Instance-to-class distance focuses on part similarity, which can add the robustness in classifier. At the same time, it neglects the global information like that in histogram, which directly encodes the overall distribution of features.

NBNN-kernel is to kernelize the NBNN classifier, and the core ideas underlying the NBNN algorithm are preserved and can be combined with the mature technology of kernel-based learning. The kernel of two sets of descriptors is as:

$$\begin{aligned} K(D_i, D_q) &= \sum_{c \in C} K^c(D_i, D_q) \\ &= \frac{1}{|D_i||D_q|} \sum_{c \in C} \sum_{d \in D_i} \sum_{d' \in D_q} k^c(d, d') \end{aligned} \tag{11}$$

Where

$$k^c(d, d') = f(dis_d^{\,1}, dis_d^{\,2}, ... dis_d^{\,|C|})^T \, f(dis_d^{\,1}, dis_d^{\,2}, ... dis_d^{\,|C|})$$

There are two choices for function f as:

Kernel 1: $$f(dis_d^{\,1}, dis_d^{\,2}, ... dis_d^{\,|C|}) = dis_d^{\,c} = \| d - d_{NN}^{\,c} \|^2 \tag{12}$$

Kernel 2: $$f(dis_d^{\,1}, dis_d^{\,2}, ... dis_d^{\,|C|}) = dis_d^{\,c} - dis_d^{\,\bar{c}} \tag{13}$$

Where $dis_d^{\,\bar{c}}$ represents the closest distance to all classes except c.

Kernel 1 corresponds to the average of the distances to the nearest neighbors for all features extracted from the test instance, which is very similar to the sum of distances used in the NBNN algorithm. For kernel 2, we subtract the distance to the nearest neighbor not belonging to class c. This corresponds to using the likelihood ratio instead of the likelihood in (4).

IV. EXPERIMENTAL RESULT

A. Descriptor Generation

Local space-time features have recently become a popular video representation for action recognition. It takes two steps, named detector stage and descriptor stage. Here, for detector, Harris detector is adopted. The hog/hof descriptors were introduced by Laptev et al. in [10]. To characterize local motion and appearance, the authors compute histograms of spatial gradient and optic flow accumulated in space-time neighborhoods of detected interest points. We use the executable code from the authors' website[1] and apply their recommended parametric settings for all detectors and descriptors.

B. KTH Corpus

The KTH dataset [7] is a commonly used benchmark action dataset with six human action classes (boxing, hand-waving, handclapping, jogging, running and walking) performed by 25 subjects in four different scenarios (outdoors, outdoors with scale variation, outdoors with different clothing, and indoors) with 2391 video samples in total. We follow the original

[1]http://www.di.ens.fr/~laptev/download.html

experimental setup [2].

TABLE I
PERFORMANCE OF KTH CORPUS BASED ON THREE KINDS OF INSTANCE-TO-CLASS CLASSIFIER

SVM	NBNN	LNBNN	NBNN-Kernel
90.232%	90.698%	93.488%	Kernel 1: 91.628% Kernel 2: 91.163%

Table I show the performances of instance-to-class based classifier. Compared with SVM with instance-to-instance distance, NBNN has about 0.466% improvements. For two kinds of kernels, NBNN-kernel can achieve better performance with 91.628% and 91.163% recognition rates for kernel 1 and kernel 2. The best result, about 93.488% is obtained by LNBNN, which is among the top rank of KTH results in all evaluation reports [12].

Fig. 3 gives the confusion networks based on KTH corpus. More details about the recognition rates can be seen from these figures. For these six actions, running is easy to be confused with jogging, and handwaving is similar to handclapping. For SVM, 25% running action is misclassified into jogging and for jogging action, 11% is labeled for running. For LNBNN and NBNN, this situation is better. The reason maybe for LNBNN and NBNN, instance-to-class distance can be robust to the similar but different action since only part of the samples will attend the distance computing. While for SVM, all samples will take part in training stage. As for handwaving and handclapping actions, all confusion networks appear the misclassification. LNBNN is the best one and only 6% handwaving action is labeled as handclapping. In fact, the moving directions of interested points in handwaving and handclapping actions are similar, but the positions are different. If the location information is added, it will be easier to distinct these two similar actions.

C. HMDB Corpus

HMDB corpus has recently been released and contains 51 distinct categories with 6766 video clips extracted from a wide range of sources. It is the largest and perhaps most realistic dataset up until now [8]. In order to compare with KTH database, we select a subset of HMDB dataset after video stabilization, which is similar with actions in KTH, named the general body movements. In this subset, there are 19 action categories (cartwheel, flic_flac, clap, climb, climb_stairs, dive, fall_floor, handstand, jump, pullup, pushup, run, sit, situp, somersault, stand, turn, walk, wave) and 2963 clips in total.

TABLE II
PERFORMANCE OF HMDB SUBSET CORPUS BASED ON THREE KINDS OF INSTANCE-TO-CLASS CLASSIFIER

SVM	NBNN	LNBNN	NBNN-Kernel
27.241%	31.217%	35.979%	Kernel 1: 30.3351 Kernel 2: 29.9824

Table II presents the comparison of these approaches. It is obviously that instance-to-class classifiers performances are better than traditional SVM. There are about 8.738% improvement by LNBNN and 3.976% improvement by NBNN. For NBNN-kernel, the first kernel function is better than the second one. But both of these two results are better than SVM. All in all, the recognition rates on HMDB are far from actual application. The whole level is similar with the evaluation on its website[3]. For C2 feature, it is 23.0%, and for HOG/HOF descriptors in this paper, it is only 20.0%. The best result is obtained by action bank [13], which is proposed this year in CVPR. To note that the reported performances are conducted on the whole HMDB database with 51 classes. For its sub-database with 19 actions, the performance should be a little better than those, which is confirmed in Table II.

Figs. 4-8 reflect the details of each class performance by confusion network. For some classes, as flic_flac, performances of NBNN and LNBNN are much better than SVM-based approach including traditional SVM and NBNN kernel. The reason may be for this kind of class, the scatter degree is much bigger which deduces the representative samples in SVM are not stable. On the contrary, for NBNN and LNBNN as long as there is one instance similar with the test one, it would be correctly labeled. But for some classes, as turn, since there are no correspondence among the instances in this class, NBNN and LNBNN's performances are worse than SVM-based classifiers. It is not strange to note this, because SVM can find a compromise classification plane no matter how bad the samples are.

[2]Training data set is as [1, 4, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 23, 24, 25]. Testing data set is as [2, 3, 5, 6, 7, 8, 9, 10, 22]

[3]http://serre-lab.clps.brown.edu/resources/HMDB/eval/

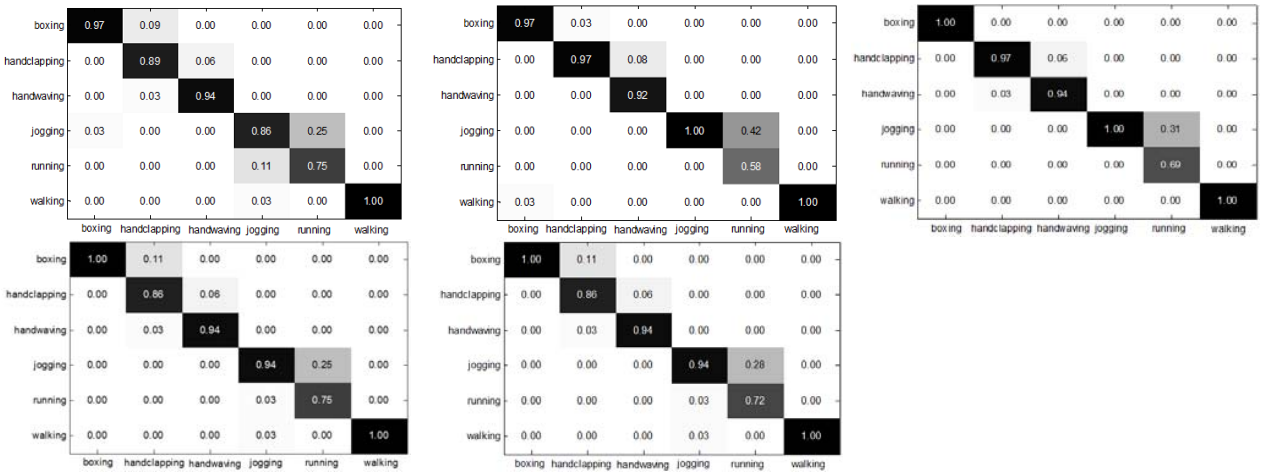

Fig. 3 Confusion networks of SVM, NBNN, LNBNN, NBNN_Kernel 1 and NBNN_kernel 2 of KTH corpus (from left to right and from top to down)

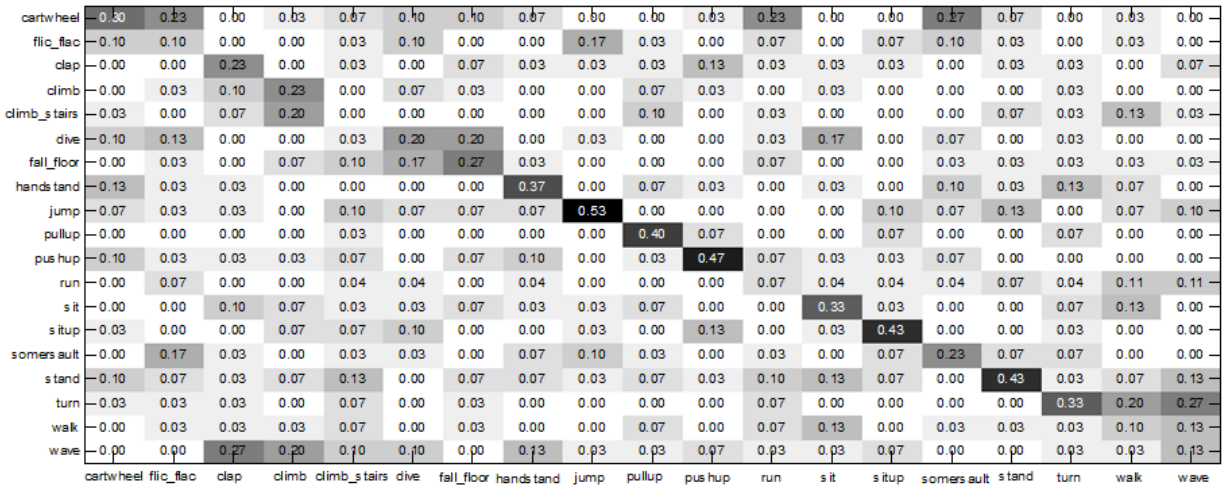

Fig. 4 Confusion networks of SVM with 3000 vocabulary size on HMDB sub-dataset (19 classes)

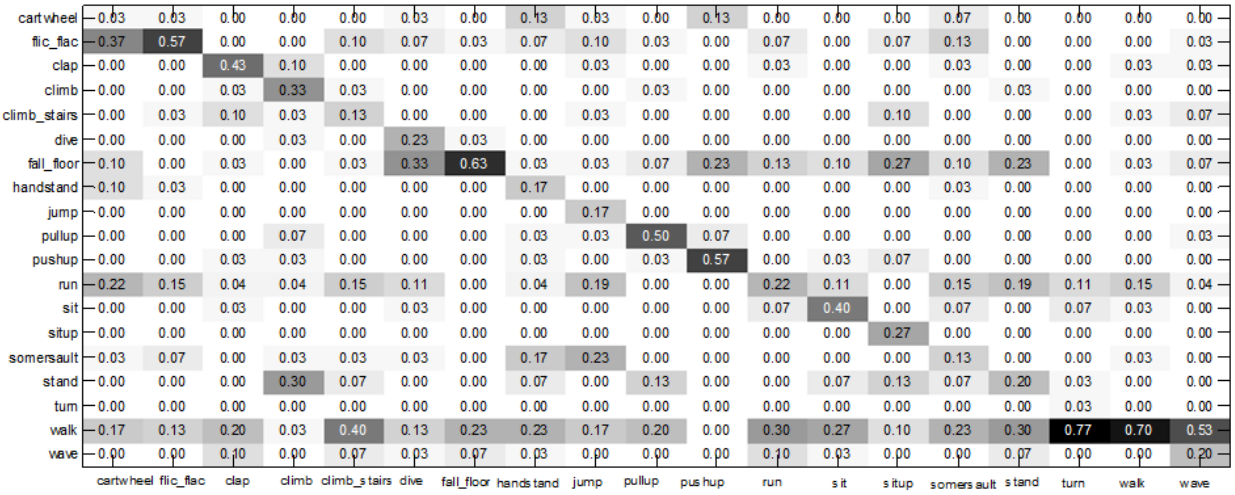

Fig. 5 Confusion networks of NBNN on HMDB sub-dataset (19 classes)

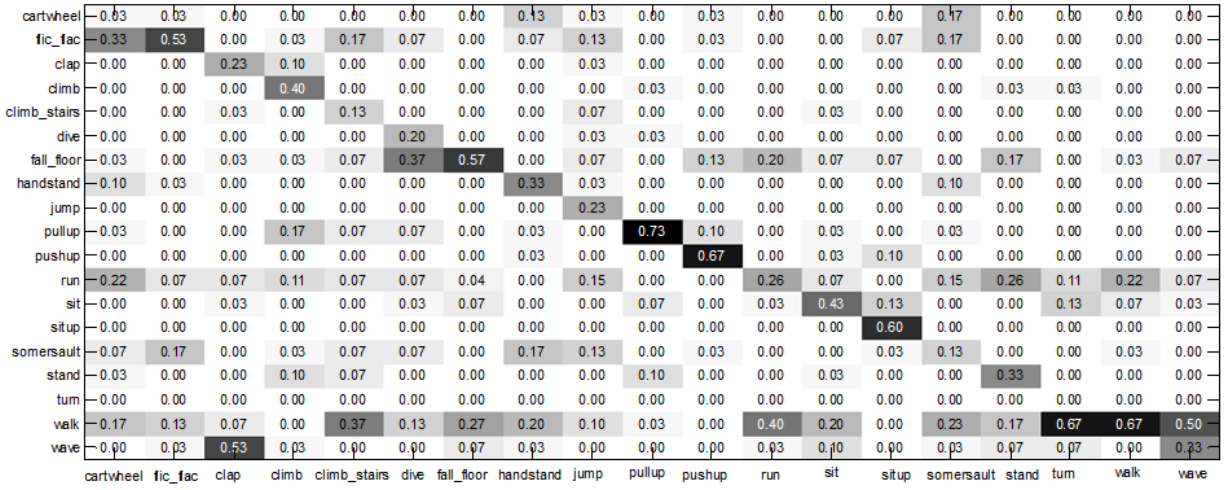

Fig. 6 Confusion networks of LNBNN on HMDB sub-dataset (19 classes)

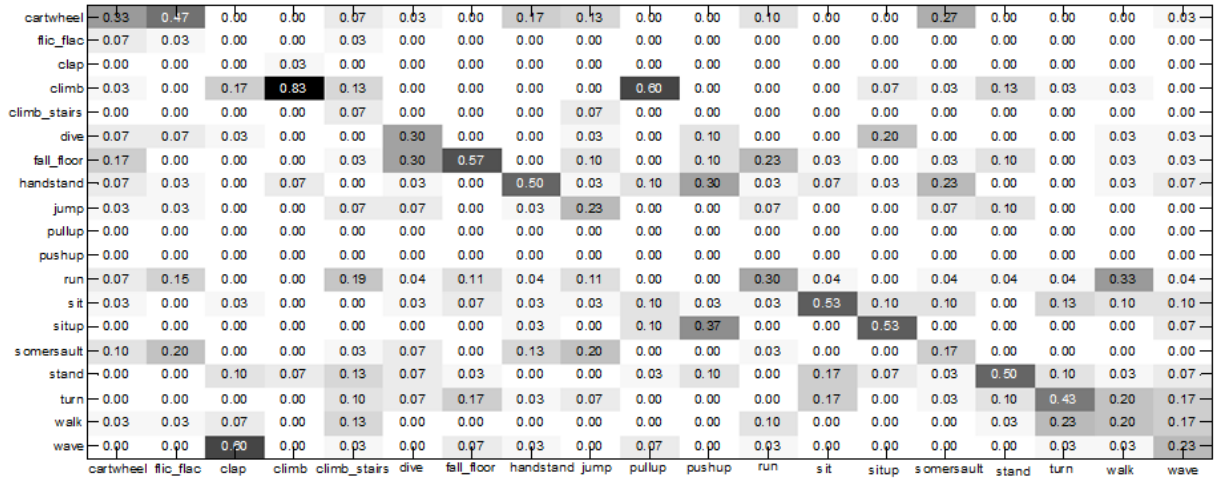

Fig. 7 Confusion networks of NBNN kernel with the first kernel function on HMDB sub-dataset (19 classes)

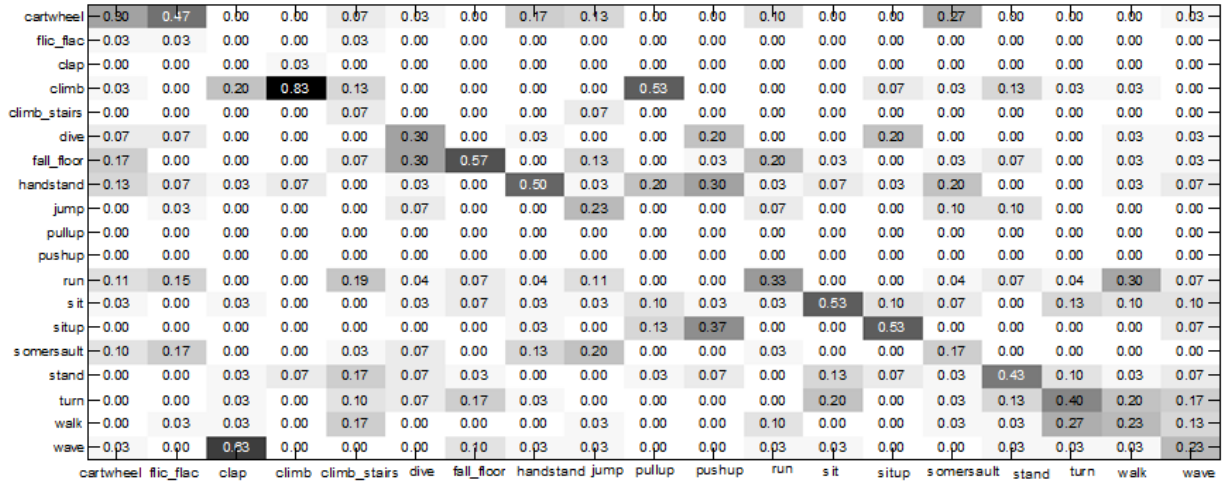

Fig. 8 Confusion networks of NBNN kernel with the second kernel function on HMDB sub-dataset (19 classes)

V. Conclusion

This paper discussed the performance of instance-to-class based classifiers, including NBNN, Local NBNN and the NBNN kernels for KTH and HMDB database. From the recognition rates, the best classifier is LNBNN, and then the NBNN and NBNN kernel. These three approaches are much better than traditional SVM from Table I and Table II. But from confusion network, especially for HMDB, it can be seen that not for all classes, LNBNN is better than SVM. For some classes, in which the samples between training ones and testing ones have no correspondence, the global strategy that all samples are candidates in training classifier are important.

Acknowledgment

This work is supported by Young Teacher Support Plan by Heilongjiang Province and Harbin Engineering University in China (No.1155G17), and Fundamental Research Funds for the Central Universities in China.

References

[1] Oren Boiman, Eli Shechtman, Michal Irani. “In Defense of Nearest-Neighbor Based Image Classification”. In CVPR 2008.
[2] T. Tuytelaars, M. Fritz, K. Saenko, T. Darrell. “The NBNN kernel”. In ICCV 2011.
[3] S. Gao, I. Tsang, L. Chia, and P. Zhao. “Local features are not lonely–Laplacian sparse coding for image classification”. In CVPR, 2010.
[4] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. “Locality -constrained linear coding for image classification”. In CVPR, 2010.
[5] Lingqiao Liu, Lei Wang, Xinwang Liu. “In Defense of Soft-assignment Coding”. In ICCV 2011.
[6] Sancho McCann, David G. Lowe. “Local Naive Bayes nearest Neighbor for Image Classification”. Technical Report TR-2011-11, University of British Columbia.
[7] C. Schuldt, I. Laptev, and B. Caputo, “Recognizing human actions: a local SVM approach,” in ICPR, 2004.
[8] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre. “HMDB: A Large Video Database for Human Motion Recognition”. In ICCV 2011.
[9] David G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. International Journal of Computer Vision 60(2), 91–110, 2004.
[10] I. Laptev, “On space-time interest points,” IJCV, vol. 64, no. 2, pp. 107–123, 2005.
[11] Zhengxiang Wang, Yiqun Hu, Liang-Tien Chia. “Image-to-Class Distance Metric Learning for Image Classification”, In ECCV 2010.
[12] Heng Wang, Muhammad Muneeb Ullah, Alexander Kläser, Ivan Laptev, Cordelia Schmid. “Evaluation of local spatio-temporal features for action recognition”. In BMVC 2009.
[13] Sadanand S., Corso J. “Action bank : a high level representation of activity in video”. In CVPR 2012.