Journal of Advances in Computer Research
Quarterly pISSN: 2345-606x eISSN: 2345-6078
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 7, No. 1, February 2016), Pages: 117-130
www.jacr.iausari.ac.ir

Human Action Recognition Based on Discriminative Sparse Representation on Multi-Manifolds

Atefe Aghaei✉[1], Sajjad Tavassoli[2]

1) University college of Rouzbahan, Sari ,Iran
2) Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari ,Iran
aghayiatefe@yahoo.com; tavassoli_5@yahoo.com

Received: 2015/03/18; Accepted: 2015/08/08

Abstract

Human action recognition is an important problem in computer vision. One of the methods that are recently used is sparse coding. Conventional sparse coding algorithms learn dictionaries and codes in an unsupervised manner and neglect class information that is available in the training set. But in this paper for solving this problem, we use a discriminative sparse code based on multi-manifolds. We divide labeled data samples into multi-manifolds and also to decrease run time, reduce dimension of manifolds. We find k inter nearest neighbors and intra nearest neighbors for each data sample in each manifold. The intra class variance should be minimized while the inter class variance should be maximized, in the result we could calculate laplacian matrix and optimize sparse code and dictionary. Then we use discriminative sparse error for classification. We run this method on KTH and UCF sport datasets. Results show that we obtain a better result (about 89%) in UCF dataset.

Keywords: *Action recognition, Discriminative Sparse Representation, Multi Manifold, Spatio-Temporal descriptors, Neighborhood*

1. Introduction

Human action recognition is the process of labeling video sequences consists of human action by its related class. Human action recognition is an important and challenging problem in computer vision. It has an extensive variety of uses, such as video content analyses, activity surveillance and human-computer interaction [1]. In general, feature representation can be divided into two categories [2]: global representation [3-5] and local representation [6-9]. Global representations are obtained in a top-down fashion: a person is localized first in the image using a background subtraction or tracking. Then the locale of interest is encoded overall, which results in the image descriptor. This representation is capable since they encode a significant part of the data. However, they depend on accurate localization, background subtraction or tracking. Additionally, they are more sensitive to viewpoint, noise and occlusions. When the domain considers the great control of these factors, global representation usually performs well. Local representations, utilizing a bottom-top procedure, are based on the spatio-temporal interest points. Without the need to subtract the background or track, these representations describe a video as a collection point changes, noise, appearance and partial occlusions. Local representation divided into three sections: first interest points are extracted from each frame, second, spatio-temporal descriptors are calculated, and after that with classification, each video sequence is assigned to the proper action class.

In recent years, a variety of spatio-temporal detectors and descriptors have been presented. There are different spatio-temporal interest point detectors including Harris 3D detector [10], Hessian detector [9] and dense sampling detector, and different descriptors including Cuboid descriptor [6], HOG/HOF descriptor [7], HOG3D descriptor [8], and extended SURF [9]. In this paper we utilize an approach introduced in [11] that is Local Motion Pattern (LMP). It was noted in [6] that the extreme points are often located in the region having spatially recognized structure. This approach is detecting the spatially distinctive points and then catches the temporal changes in the neighborhood of those points.

Numerous classifications including k nearest neighbors [12], Support Vector Machine (SVM) classification and sparse coding [11] are introduced. Sparse coding (Sc) is strong tool for analyzing signals such as voice, image and video [13, 14, 15] including face recognition, speech recognition, handwritten digit recognition, image clustering, etc. Given a set of data features (from the description) as an input data matrix, sparse representation aims to find a sparse matrix using this input data and an over complete codebook that got from descriptors. Because of the over complete codebook which is learned by Sc, the locality of the data samples to be encoded may be disregarded. As a result similar data vectors may be represented as sparse codes. So Graph-regularization Sparse Coding (GraphSc) and Laplacian Sparse coding (LSc) were proposed by Zheng et al. [16] and Gao et al. [17, 18] respectively. In both methods, the local geometrical structure of the dataset is explicitly explored by building a P-nearest neighbor graph, and the graph Laplacian is used as a smooth operator to preserve the local manifold structure.

Most of sparse coding algorithms learn dictionaries and codes in an unsupervised manner and so neglect class information that is available in training set. In this article, we use a supervised sparse coding method that dictionaries and sparse codes learn from both of datasets and class labels. In order to decrease run time, dimension of the descriptors that are extracted from datasets is reduced. Then descriptors are divided into some manifolds. In this method we find k inter-nearest neighbors and intra-nearest neighbors for each data sample in each manifold. The intra class variance should be minimized while the inter class variance should be maximized. In order to acquire to this aim, we calculate intra class and inter class weighted function. After that, we calculate laplacian matrix with using these functions. Then we optimize sparse cod and dictionary with using laplacian matrixes. For classification we use a novel approach called discriminative sparse code error. This method uses a linear transformation matrix that is optimized using optimized sparse code matrix. Proposed method has two main advantages: been discriminative is one of the advantages. The second advantage is that, optimization of sparse code and classifications are doing simultaneously. We describe proposed approach in section 3. We introduce two algorithms as training algorithm and test algorithm in section 4. Also section 5, demonstrate experimental results in simple dataset like KTH and complicated dataset like UCF sport. Our approach has better result than other approaches in complicated datasets.

2. Literature Review

Many researchers have been studied Human recognition .For instant Yilmaz et al. [20] and Fanti et al. [21], define an approach that was based on tracking human body parts and use the derived motion trajectories, to perform action recognition. Liu et al. [22] use maximum mutual information to learn code book and use space-time pyramid, to exploitation of temporal information. In [25] an approach is used that is, first, according to the geometrical features of the human body, organs are recognized and then human pose is estimated. Fathi et al. [26] extract discriminative flow features within overlapping space-time cells and select mid-level features via Adaboost. Wang et al [23] use the hidden conditional random fields for action recognition. They model a class of actions as a root template, and establish an association of hidden parts which are a group of local patches that are related to intermediate representations. Malgireddy et al. [27] propose to extract local features (Histogram of Flow and Histogram of Oriented Gradient) on each frame and apply a bag of-words step to obtain a global description of the frame. This action is then modeled as a multi channel Hidden Markov Model.

The most similar attempts to this paper are researches that are based on sparse representation. For instance Guo.K et al. [38] propose a method using empirical covariance matrices of bags of low dimensional feature vectors. They use sparse linear approximation of a query vector in an over complete dictionary of training vectors. Guha et al. [11] use traditional sparse code and in order to model human actions, three over complete dictionary learning framework are investigated. An over complete dictionary is developed using a set of spatio-temporal descriptors (extracted from the video sequences) in such a way that each descriptor is represented by some linear combination of a small number of dictionary elements. Gao et al. [28] propose an algorithm based on sparse representation induced by L1 and L2 regulations called SRL12.Wang et al's research [37] propose another spatio-temporal descriptor named Locally Weighted Word Context. They use graph regularized non negative matrix factorization to learn action unit. Also for classification, they use sparse representation based on joint $L_{2,\ 1}$-norm. Zheng etal.[37] propose a new method for human action recognition called Local Manifold-Constrained that represent an interest point by several words, which lie in the same manifold as the interest point does. Also they use Manifold-constrained term to improve accuracy in classification and incorporate it into the objective function of SRC. The method that we propose for human action recognition has a new term in objective function called discriminative sparse representation and incorporates it into objective function of SRC and large margin term that is proposed in [13].

3. Proposed approach

This approach consists of four stages: 1. Computation of spatio-temporal motion descriptors 2. Dimension reduction 3.Optimizing the sparse codes by using descriptors and initial dictionary, and update dictionary for each class 4.Classification of unlabeled data using dictionaries, corresponding sparse codes and a linear transformation matrix.

We have shown our approach that we use in this paper in figure 1.

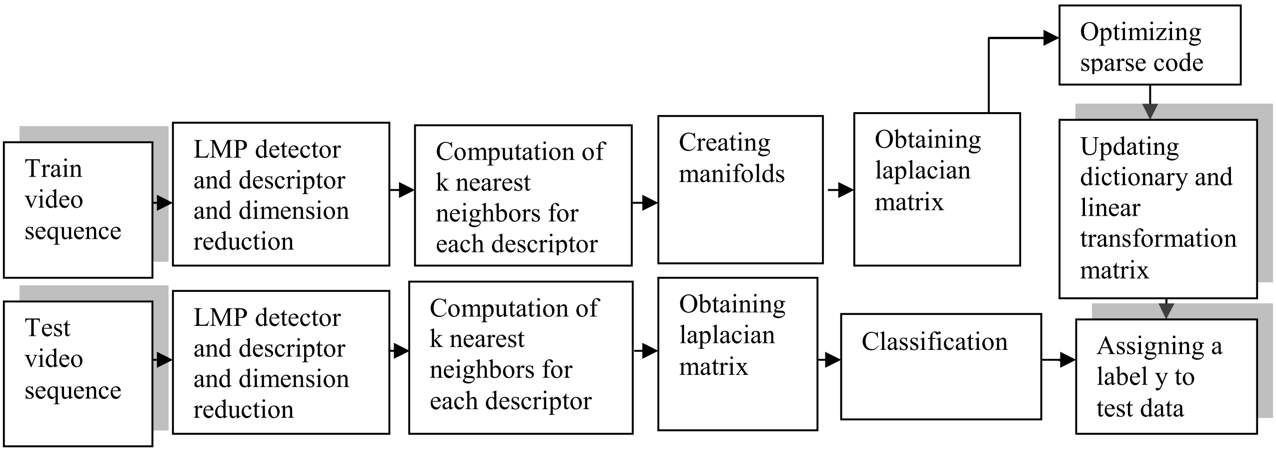
Train video sequence
LMP detector and descriptor and dimension reduction
Computation of k nearest neighbors for each descriptor
Creating manifolds
Obtaining laplacian matrix
Optimizing sparse code
Updating dictionary and linear transformation matrix
Test video sequence
LMP detector and descriptor and dimension reduction
Computation of k nearest neighbors for each descriptor
Obtaining laplacian matrix
Classification
Assigning a label y to test data

Figure 1: flowchart of the proposed approach

3.1 Computing spatio-temporal descriptors

In this section we extract spatio-temporal descriptor for every video sequence. In some previous researches, the extreme points are frequently found in the districts which having spatially recognized structure. Hence we choose to use Local Motion Pattern [11]. According to this method, first spatially different points would be identified and then we catch temporal changes in the area of those points.

Detector: assume that we have a video sequence contain of "F" frames. First, each video sequence divided into "S" segments, so as each segment includes c=F/S consecutive frames. Then we perform a detector using Harris 2D detector [29], and this detector compute key points of the first frame of each segment, then we obtain temporal changes for the rest (c-1) of remaining frames of that segment. So for "c" frames of each segment, size of these key points is $(n\times n\times c)$.

Descriptor: Each key point is a spatio-temporal cube of size *$(n\times n\times c)$*. Each cube captures local space-time changes and represents a significant motion pattern. To obtain a robustness descriptor, first, we perform a 2D Gaussian blurring in the special domain to ignore minor variations. Note that this Gaussian blurring should not be in the temporal domain, because, we want to capture temporal changes, so these changes should not be ruined.

We remove mean of that cube, then compute variance (M_2), skewness (M_3) and kurtosis (M_4). We use a moment matrix Mr, that is defined in [11], r= {2, 3, and 4} associated with v as (1):

$$M_r = [mij],\ i, j = 1, 2, \ldots, n.$$

$$mij = 1/l\sum_{t=1}^{l}(vij)^r \qquad \textbf{(1)}$$

Where v_{ijt} is the value pixel of "t" th patch at location {i,j}. These three value vectors are connected on top of each other to form a vector $m\in R^d$ (d=3n^2).

3.2 Dimensionality Reduction

If the size of patches (n×n×c) is equal to (24×24×c) then dimension of LMP become (1728×1). In order to create an over complete dictionary $D \in R^{n\times m}$ (m>>2n) we required more than 3500 atoms for every vector. So as to tackle this issue, we reduce dimensions of descriptors. The method that we use is Random Projection (RP) [32] which is simple and quick.

We have a set of “P” descriptors from a video sequence and dimension of each descriptors is d. this set can be represented by v $\in R^{d\times p}$. This matrix is projected onto an n-dimensional subspace (n<<d) by premultiplying the descriptor matrix “v” by a random matrix R $\in R^{n\times d}$. So we obtain a matrix X $\in R^{n\times p}$:

$$X = RD \qquad \textbf{(2)}$$

3.3 Sparse Representation

We assume that the descriptors which were calculated are $X=\{x_i\}=\{x_1,x_2,\ldots,x_p\}\in R^n$. So that, p is the number of data samples and n is the dimensionality of feature vectors. These data according to their labels are divided into L classes. So $x_L=\{x_i \mid y_i=L, x_i \in X\}$. X_L is the data which is in Lth class that we represent it by a manifold (M_L).

3.3.1 Discriminative sparse code and sparse code error on multi-manifold

In this method we set a dictionary matrix $D=\{d_{i1},d_{i2},\ldots,d_{ik}\}$ for each manifold. In order to build initial dictionary, we choose some data of each manifold randomly. Also S_{Li} is coefficient vector of $x_i \in x_L$ which is sparse coding of this data sample. Here we have a linear combination that is $X_i = \mathrm{D}_L S_{Li}$.

This sparse code should minimize loss function (3)

$$\min_{D_L,S_L}\left\{F(D_L,S_L)=\sum_{Ex_L\in X_L}\left(\left\|x_i-D_L s_{Li}\right\|^2+\alpha\left\|s_{Li}\right\|_1\right)\right\} \qquad \textbf{(3)}$$

$$s.t.\left\|d_{Lk}\right\|^2\le c,\quad k=1,\ldots,k,$$

Where $\left\|S_L\right\|$ is l_1-norm function to measure the sparseness of S_L .Also it should keep the reconstruction coefficient as sparse as possible.

Consider that $\{x_1 \ldots x_p\}$ are our data samples belonging to L manifolds; we define two kinds of neighbors, Intra-class $\left(N^{Intra}\right)$ and Inter-class$\left(N^{inter}\right)$. Intra-class nearest neighbors of a data sample (xi) are k nearest data samples from same class as x_i, and Inter-class neighbors are k nearest data samples from different classes from x_i.

With a usage of Gaussian kernel we are calculating the intra weighted matrix $\left(W^{Intra}\right)$ and the inter weighted matrix $\left(W^{Inter}\right)$ just like (4):

$$w_{Lij}^{\mathrm{int}\,ra}=\begin{cases}\exp(-\dfrac{\left\|x_i-x_j\right\|^2}{2\sigma^2}), if\; x_i\in X_l, and(x_j\in N_i^{\mathrm{int}\,ra}\, or\; x_i\in N_j^{\mathrm{int}\,ra})\\ 0\end{cases}$$

$$w_{Lij}^{\mathrm{int}\,er}=\begin{cases}\exp(-\dfrac{\left\|x_i-x_j\right\|^2}{2\sigma^2}), if\; x_i\in X_l, and(x_j\in N_i^{\mathrm{int}\,er}\, or\; x_i\in N_j^{\mathrm{int}\,er})\\ 0\end{cases} \qquad \textbf{(4)}$$

Minimizing optimization problem (4) with using weighted functions [30] in equation (5):

$$\min_{s_L}\left\{\begin{array}{l} M\left(S_L\right)=\frac{1}{2}\sum_{i:x_i\in X_L}\left(\sum_{j:x_j\in N_L^{\mathrm{int}\,ra}}\left\|s_{Li}-s_{ij}\right\|^2 W_{Lij}^{\mathrm{int}\,ra}\right) \\ \qquad -\frac{1}{2}\sum_{i:x_L\in X_L}\left(\sum_{j:x_j\in N_L^{\mathrm{int}\,ra}}\left\|s_{Li}-s_{ij}\right\|^2 W_{Lij}^{\mathrm{int}\,ra}\right)\end{array}\right\} \qquad \textbf{(5)}$$

According to this function, Intra class variance should be minimize and inter class variance should be maximize. In other word, the data samples that are in intra-class must get closer, and the data samples that are in inter-class must be as far as possible.

Additionally, we consider another term that is “discriminative sparse code error”. Assume that we have a matrix $Y\in R^{l\times n}$.so that, in each column all of the components are zero except one of them which is 1. The Discriminative sparse error term is (6):

$$\left\|y-As\right\|_2^2 \qquad \textbf{(6)}$$

Where $A\in R^{l\times d}$ is a linear transformation matrix.

In order to write objective function, we should consider three terms that we have introduced: the sparse reconstruction lose term (3), the large margin term (5) and sparse code error term (6).

Consequently, objective function is (7):

$$O\left(D_L,S_L,A\right)=Min\left\|x_L-D_LS_L\right\|^2+\alpha\sum_x^{nl}\left\|s_{L\mathrm{n}}\right\|_1+\beta Tr\left(S_LL_LS_L{}^T\right)+\left\|y_L-AS_L\right\|_2^2 \qquad \textbf{(7)}$$

Where $L_L=L_L^{intra}-L_L^{\mathrm{int}\,er}$ is laplacian matrix related to each class, so that $L_L^{intra}=Di_L^{intra}-W_L^{intra}$ and $L_L^{inter}=Di_L^{inter}-W_L^{inter}$, Di^{intra} and Di^{inter} are diagonal matrixes of intra-class and inter-class respectively, which are calculated as: $Di_L^{intra}=\sum_{m=1}^{nL}w_{L\mathrm{n}m}^{\mathrm{int}\,ra}$ and $Di_L^{intra}=\sum_{m=1}^{nL}w_{L\mathrm{n}m}^{\mathrm{int}\,er}$, and β is trade off parameter. We aim to minimize $O\left(D_L,S_L,A\right)$.

3.4 Optimization

According to this method, optimization is accomplished in three steps:

1- To Optimize sparse code
2- To Update dictionary
3- To Update linear transformation matrix

That we are going to explain these steps separately.

3.4.1 Optimizing sparse code

In this step, first, we should set initial values for sparse codes, dictionaries and linear transformation matrix. For sparse code matrix, we set a zero matrix as initial value. In order to build initial dictionary we choose some data of each manifold randomly. Also in order to build initial transformation matrix we set a random matrix.

We acquire sparse code under one condition that is the dictionary (D_l) should be fixed. It means that, in step 1, we use the initial dictionary. So we have (8):

$$Min\left\|x_L-D_LS_L\right\|^2+\alpha\sum_x^{nl}\left\|s_{Ln}\right\|_1+\ \beta Tr\left(S_LL_LS_L{}^T\right) \qquad \textbf{(8)}$$

Consider that multi-manifold laplacian regularizer [13] can be written as:

$$Tr\left(S_LL_LS_L^t\right)=\sum_{n,m-1}^{n_L}L_{nm}s_{L\mathrm{n}}^Ts_{lm} \qquad \textbf{(9)}$$

Then (9) rewritten as equation (10):

$$\min_{s_{L\mathrm{n}}} \left\| x_n - D_L S_{L\mathrm{n}} \right\|^2 + \alpha \left\| s_{L\mathrm{n}} \right\|_1 + \beta \left[L_{nn} s_{L\mathrm{n}}^T s_{L\mathrm{n}} + 2 s_{L\mathrm{n}}^T L_{nm} s_{Lm} \right] \tag{10}$$

This problem can be optimized by the graph regularized sparse code algorithm that is introduced in [13].

3.4.2 Updating dictionary

In this step, in order to update dictionary, the sparse code which is calculated in previous section, should be fixed. So optimization problem is:

$$\begin{aligned} &\min_{D_L} \left\| X_L - D_L S_L \right\|^2 \\ &s.t. \left\| d_{Lk} \right\|^2 \le c,\ k = 1, \ldots, k. \end{aligned} \tag{11}$$

There are many methods such as gradient descent with iterative projection to solve this problem. In this paper we adopt the method that uses a Lagrange dual which has been shown more efficient than gradient descent. So the optimal solution D* can obtain as (12) that is introduced in [13]:

$$D_L^* = X_L S_L^T \left(S_L S_L^T + diag(\lambda^*) \right)^{-1}, \tag{12}$$

Where $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_k]$ is the multiplier [34] related to kth inequality constraint $\left\| d_{L\mathrm{k}} \right\|^2 \le C$, and λ* is the optimal solution of λ [35].For more information refer to [13].

3.4.3 Updating linear transformation matrix

In this section, we update linear transformation (known as A) by fixing discriminative sparse code “s” that is optimized in previous section. The optimal solution “A" can be obtained by letting the first derivative of (6), equal to zero. So we have equations (13) and (14):

$$2(Y_L - A s_L) {s_L}^T = 0 \tag{13}$$

$$A = Y_L {s_L}^T (s_L {s_L}^T)^{-1} \tag{14}$$

3.5 Classification

When we want to test some data(x$_t$) we assume there is a loop and each time x$_t$ belongs to one of the “L” manifolds, so we calculate intra-class and inter-class nearest neighbors for data samples of the manifold, then we calculate W^{intra} and W^{inter} using (4).

After that we compute corresponding laplacian matrix, finally we optimize sparse code for test data with (15)[30] and compute “y” matrix for each class with (16).Function (15) only update sparse codes for test data sample with keeping spares code of $x_n \text{Є} M_L$ fixed. And also in this function the dictionary of each class (*D$_L$)* is fixed.

$$\min_{s_t} \left\{ \left\| x_t - D_L s_{Lt} \right\|^2 + \alpha \left\| s_{Lt} \right\|_1 + \frac{\beta}{2} \left(\sum_{nx_a \in N_{Lu}^v} \left\| s_{Lt} - s_{L\mathrm{n}} \right\|^2 W_{Lm}^{\mathrm{int}ra} - \sum_{nx_a \in N_{Lu}^v} \left\| s_{Lt} - s_{L\mathrm{n}} \right\|^2 W_{Lm}^{\mathrm{int}ra} \right) \right\} \tag{15}$$

$$Y_i = A s_i \tag{16}$$

At the end of the algorithm, we assign a label *s$_t$* to *x$_t$*like (17):

$$y_t \leftarrow \arg\max y_i. \tag{17}$$

4. Algorithms

The algorithms that are used in this paper are summarized in algorithm 1 and algorithm 2 that are Train algorithm and Test algorithm.

```
Algorithm 1: Train
Input: labeled sequence videos
Step1:
Computation of LMP detectors and descriptors.
Step 2:
The descriptors computed at step1, are divided to L manifolds (according to data labels)
Reduce dimension of each manifold using RP.
Step 3:
For i=1 to L
Compute k intra nearest neighbors and inter nearest neighbors of data samples in ith manifold
Compute weighted matrix using equation (4)
Compute laplacian matrix corresponding to ith manifold
Compute initial code book, sparse code and linear transformation matrix.
Step 4:
Optimizing sparse code using (10)
Updating dictionary using (11)
Updating linear transformation matrix using (14)
```

```
Algorithm 2: Test
Input: test video sequence
Step1:
Computation of LMP detector and descriptor
Step 2:
For i=1 to L
Add test data into the ith manifold
Compute k intra and inter nearest neighbors for data samples in ith manifold
Compute weighted matrix using equation(4)
Compute laplacian matrix corresponding to ith manifold
Step 3:
Optimize sparse code for test data samples using (15)
Step 4:
Classification using (16)
Assigning a label $y_t$ to test data using (17)
```

5. Experimental results

In this section we present the results that are obtained from our method.

5.1 Evaluation method

We run proposed method on two datasets: KTH and UCF Sport. In order to obtain an appropriate accuracy of this method, we use *K-fold* cross validation. In this validation method, the dataset is divided into *K* subsets, and the holdout method is repeated *K* times. Each time, one of the *K* subsets is used as the test set and the other *K-1* subsets are put together to form a training set. Then the average error across all *K* trials is computed. In this paper we use 10-fold cross validation.

5.2 Datasets

In this section we introduce the dataset that are used in our approach. The datasets that are investigated in this paper are a simple dataset, named KTH dataset and a complex dataset named UCF sport.

5.2.1 KTH Dataset

This video dataset, containing six types of human actions: walking, jogging, running, boxing, hand waving, and hand clapping, performed several times by 25 persons in four different scenarios: indoors s1, outdoors s2, outdoor with different clothes s3, and outdoors with scale variation s4. There were total 2391 sequences taken with a static camera. The frame size was 160×120.

The parameter $\boldsymbol{\alpha}$ in equation (8) is sparsity parameter. We want to peruse the effect of the value change in this parameter on the average recognition accuracy on KTH dataset. We choose 6 different values for this parameter. These values change in 0.01 to 1000. Figure 2 demonstrates the corresponding results. Results show that the average recognition increases from 0.01 to 100 and then decreases. So the best value for this parameter in our paper is 100.

Also, in order to perceive the effect of the neighborhood on recognition, we set 4 different values for the number of nearest neighbors (k). We run our method for all of these values on KTH dataset and put their accuracy in figure 3. Results show that in our method, the best value for k between these values is 6.

Additionally, we consider the effect of the number of dictionary atoms on the recognition performance. 5 different dictionary sizes are evaluated on KTH dataset that are 200, 400, 600, 800, and 1000. As you can see in figure 4, the average recognition increases from the size 200 to 600, after that the recognition is fixed nearly. So the best value that we choose for the number of dictionary atoms is 600.

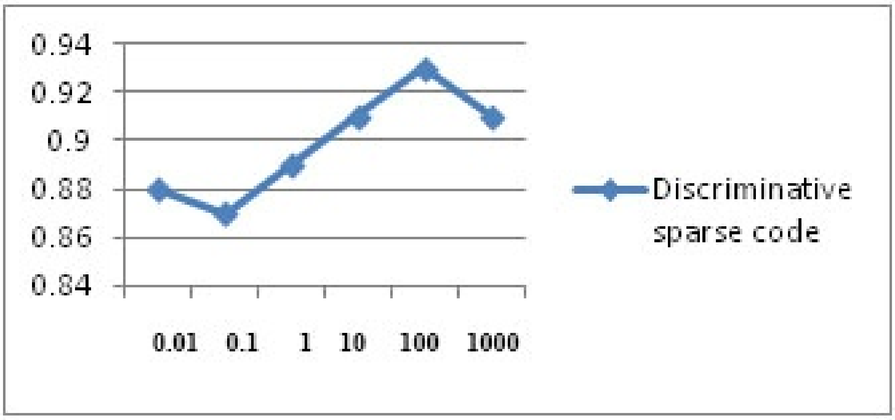

Figure 2: Discriminative sparse code and sparse error performance with different values of the sparsity parameter on KTH dataset

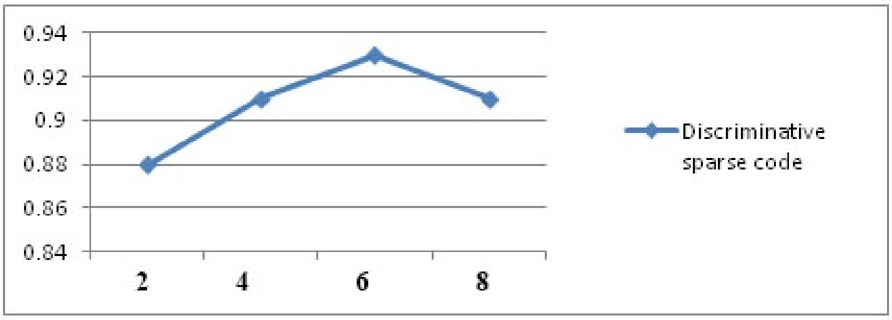

Figure 3: Discriminative sparse code and sparse error performance with different values of the number of nearest neighbors on KTH dataset

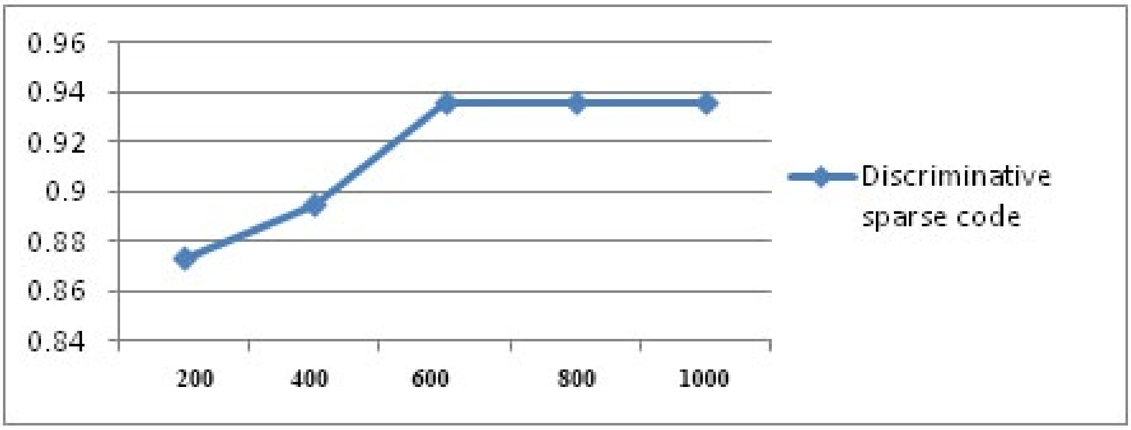

Figure 4: Discriminative sparse code and sparse error performance with different size of dictionaries on KTH dataset

Finally, we put the best results that we get from these values on KTH dataset in table 1.So experimental results on KTH dataset including 6 nearest neighbors, the best sparsity value that is calculated 100 and dictionary size 600 are in table 1. These results are obtained using matlab 2012. Due to these results boxing and jogging achieve the best results. Also we put comparison between our method and some others in table 2. Our approach has a better result just for complicated datasets which is a lot better because complicated databases are closer to the reality. As you can see in table 2, our approach has good result (93.66%), but it has less accuracy than Wang approach for KTH dataset, while has greater result in UCF sport dataset which much more intricate than KTH datasets .The reason of getting poor result in simple dataset is that LMP descriptor that we use, extract a few number of descriptors for simple datasets.

Table 1: Average accuracy on KTH Dataset

	walking	Running	jogging	boxing	Hand waving	Hand clapping
walking	0.93	0	0.07	0	0	0
running	0.01	0.89	0.10	0	0	0
jogging	0	0	1	0	0	0
boxing	0	0	0	1	0	0
Hand waving	0	0	0	0.11	0.89	0
Hand clapping	0	0	0	0.08	0.01	0.91

Table 2: Comparison between our method and some others on KTH Dataset

method	**year**	**accuracy**
Laptev et al. [7]	**2008**	**91/8**
Liu et al. [28]	**2012**	**92**
Wang et al.[36]	**2013**	**94/2**
Our approach	**2015**	**93/66**

5.2.2 UCF Sport Dataset

This dataset consists of a set of actions collected from various sports which are typically featured on broadcast television channels such as the BBC and ESPN. The video sequences were obtained from a wide range of stock footage websites including BBC Motion gallery, and Getty Images. This data set contains close to 200 video sequences at a resolution of 720x480. The collection represents a natural pool of actions featured in a wide range of scenes and viewpoints. By releasing the data set we hope to encourage further research into this class of action recognition in unconstrained environments. Actions in this data set include: Diving (16 videos), Golf swinging (25 videos), Kicking (25 videos), Lifting (15 videos), Horseback riding (14 videos), Running (15 videos), Skating (15 videos), Swinging (35 videos), Walking (22 videos).

We run our method with using the best parameters that are achieved from KTH dataset for UCF Sport dataset. So, we use 6 nearest neighbors, the best value for sparsity parameter that is 100 and dictionary size 600. Experimental results of this dataset are in table 3. As you can see in table 3 we get perfect results in UCF sport dataset, also Diving and Horseback riding achieve the best results. Also we put comparison between our method and some others in table 4. Table 4 shows that our method has better results in UCF Sport dataset (88.89%) because UCF sport dataset is a complicated dataset and our method gets better results for complicated datasets.

Table 3: Average accuracy on UCF Dataset

	Diving	Golf swinging	Kicking	Lifting	Horseback riding	Running	Skating	Swinging	Walking
Diving	1	0	0	0	0	0	0	0	0
Golf swinging	0	0.85	0.12	0	0	0.03	0	0	0
Kicking	0	0	0.84	0	0	0.06	0	0	0.10
Lifting	0	0	0	0.91	0	0	0	0.09	0
Horseback riding	0	0	0.06	0	0.89	0	0	0.05	0
Running	0	0	0	0	0	0.84	0.08	0	0.08
Skating	0	0.03	0.07	0	0	0	0.90	0	0
Swinging	0	0	0.04	0	0.06	0	0	0.90	0
Walking	0	0	0	0	0	0.08	0.05	0	0.87

Table 4: Comparison between our method and some others on UCF Sport Dataset

method	**year**	**accuracy**
Guha et al.[11]	**2012**	**83.8**
Wang et al.[36]	**2013**	**88**
Zhang.X et al. [37]	**2013**	**84.7**
Wang et al. [31]	**2014**	**88.7**
Our approach	**2015**	**88.89**

6. Conclusion and future works

This paper was about human action recognition based on discriminative sparse representation on multi-manifold. First, we divided data samples into multi manifolds, then found k intra and inter nearest neighbors of data samples in each manifold, after that we calculate laplacian matrix, related to each manifold for optimizing dictionaries and sparse codes. We used a new method called discriminative sparse code error for classification.

In order to diminish execution time, we reduce dimension of manifolds. As you can see in table 4 for complex datasets such as UCF it provide better results (about 89%).

We propose to use multi-view descriptor as future work. In this way we can assume each local descriptor as a separate view and combine them to effectively explore the complementary nature of multiple views and hope to get better results.

References

[1] P. Turaga, R. Chellappa, V. S. Subramanian, and O. Udrea, *Machine recognition of human activities: A survey,* IEEE Trans. Video Technol. vol. 18, no. 11, 2008, pp. 1473–1488.

[2] R. Poppe, *A survey on vision-based human action recognition*, in: Image and Vision Computing, 2010, pp. 976–990.

[3] Y.Wang,K.Huang,T.Tan, *Human activity recognition based on R transform*, in: CVPR, 2007, pp.1–8.

[4] V.Kellekumpu, G.Zhao, M.Pietikainen, *Human activity recognition using a dynamic texture based method*, in: BMVC, 2008.

[5] H.Jiang, D.R. Martin, *finding actions using shape flows*, in: ECCV, 2008, pp.278-292.

[6] P.Dollar, V. Rabaud, G.Cottrell, S.Belongie, *Behavior recognition via sparse spatio-temporal features*, in: VS-PETS, pp.65-72, 2005.

[7] I.Laptev, M.Marszalek, C.Schmid, B.Rozenfeld, *Learning realistic human actions from movies*, in: CVPR, 2008, pp.1-8.

[8] A.klaser, M.Marszalek, c.Schimd, *a spatio-Temporal descriptor based on 3D-gradients*, 2008, in: BMVC.

[9] [9] G.Willems, T.Tuytelaars, L.VanGool, *An effective dense and scale-invariant spatio-temporal interest point detector*, in: ECCV, 2008, pp.650-663.

[10] L.Laptev, T.Lindeberg, *Space-time interest points*, in: IEEE, ICCV conference, 2003.

[11] T.Guha, R.Kreideh, *LearningSparse Representations for Human Action Recognition*, IEEE vol 34, NO.8, in:TPAMI,2012

[12] Zhe.Zhang, *Vision based human action Recognition: A Sparse Representation Perspective*, 2012, pp.4-19.

[13] H.Lee, A.Battle, R.Riana, A.Y.Ng, *Efficient sparse coding algorithms*, in:NIPS, 2007, pp. 801-808.

[14] J.Eggert, E.Koner, *sparse coding and nmf*, in:IEEE International Conference on Neural Networks- Conference Proceeding, vol.4, 2004, pp. 2529-2533.

[15] I.Ohiorhenuan, F.Mechler, K.Purpura, A.Schmid, Q.Hu, J.Victor, *Sparse coding and high-order correlations in fine-scale cortical networks*, Nature, 2010, pp.617-621.

[16] M.Zheng, J.Bu, C.Chen, C.Wang, L.Zhang, G.Qiu, D.Cai, *Graph regularized sparse coding for image representation*, IEEE Transactions on Image Processing, 2011, pp.1327-1336.

[17] S.Gao, I.Tsang, L-T, China, P.zhao, *Local features are not lonely-Laplacian sparse coding for image classification*, in: 2010 IEEE (CVPR), 2010, pp. 3555-3561.

[18] S.Gao I-H. Tsang,L-T. China, *Laplacian sparse coding, hyper graph laplacian sparse coding, and applications*, IEEE, 2013, pp.92-104.

[19] JolliffeI.T.Principal *Component Analysis*, Series: Springer series in Statistic, 2nd Springer, NY, XXIX, 487 p.28 illus. ISBN 978-0-387-95442-4, 2002.

[20] A.Yilmaz and M. Shah, *Recognizing human actions in videos acquired by uncalibrated moving cameras*. Images," in Proc. of the tenth IEEE International conference on computer vision, 2005.

[21] C.Fanti, L. Zelnik-Manor, and P. Perona, *Hybrid models for human motion recognition*, in Proc. of the tenth IEEE international conference on computer vision, 2005.

[22] J.Liu and M. Shah, *Learning human actions via information maximization*, in Proc. IEEE Int. Conf. CVPR, 2008, pp. 1–8.

[23] Y. Wang and G. Mori, *Max-margin hidden conditional random fields for human action recognition*, in Proc. IEEE Int. Conf. CVPR, 2009, pp. 872–879.

[24] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, *Describing objects by their attributes*, in Proc. IEEE Int. Conf. CVPR, 2009, pp. 1778–1785.

[25] M.Behrouzifar, H.ShayeghBoroujeni, N. Moghadam Charkari, K.Mozafari, *Model Based Human Pose Estimation in Multi-Camera Using Weighted Particle Filters*, Springer,2012, pp. 234-243

[26] A. Fathi and G. Mori, *Action recognition by learning mid-level motion features*, in Proc. IEEE Int. Conf. CVPR, 2008, pp. 1–8.

[27] M. R. Malgireddy,I. Inwogu, and V.Govindaraju, *A temporal Bayesian model for classifying, detecting and localizing activities in video sequences*, Computer Vision and Pattern Recognition Workshops, 2012.

[28] Z.Gao, A.Liu, H.Zhang, G.Xu, Y.Xue *Human Action Recognition based on Sparse Representation Induced by L1/L2 Regulations,* 21st(ICPR), 2012

[29] C. Schmid, R. Mohr, and C. Bauckhage, *Evaluation of Interest Point Detectors*, Int’l J. Computer Vision, vol. 37, 2000, pp. 151-172.

[30] J.J.Wang,H.Bensmail,N.Yao,X.Gao*Discriminative sparse coding on multi-manifold* in: Elsevier, Knowledge-Based Systems, 2013, pp.199–206,

[31] H.Wang, C.Yuan, W.Hu, H.Ling, W.Yang, C.Sun, *Action Recognition Using Nonnegative Action Component Representation and Sparse Basis Selection* ,IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 23, NO. 2, 2014

[32] R. Baraniuk and M. Wakin, “*Random Projections of Smooth Manifolds*,” Foundations of Computational Math., vol. 9, 2009, pp. 51-77.

[33] S. Drury, S. Loisel, *Sharp condition number estimates for the symmetric 2-lagrange multiplier method*, Lecture Notes in Computational Science and Engineering 91, 2013, pp.255–261.

[34] J. Kwak, H. Cho, S. Shin, O. Bauchau, *Improved finite element domain decomposition method using local and mixed Lagrange multipliers*, in: 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2013.

[35] S. Brogniez, C. Farhat, E. Hachem, *A high-order discontinuous Galerkin method with Lagrange multipliers for advection-diffusion problems*, Computer Methods in Applied Mechanics and Engineering 264, 2013, pp.49–66.

[36] H. Wang, A. Klaser, C.Schmid, and C.Liu, *Dense trajectories and motion boundary descriptors for action recognition*, Int. J. Compute, Vis, vol. 103, no. 1, 2013, pp. 60–79.

[37] X.Zhang, Y.Yang, L.C.Jiao and F.Dong, *Manifold-constrained coding and sparse representation for human action recognition,* Elsevier pattern recognition journal, 46. 2013. pp.1819-1831.

[38] K.Guo, P.Ishwar, J.Konrad, *Action Recognition Using Sparse Representation on Covariance Manifolds of Optical Flow,* Springer, Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance, 2010, pp.188-195.