A Tree Regularized Classifier—Exploiting Hierarchical Structure Information in Feature Vector for Human Action Recognition

Huiwu Luo, Fei Zhao, Shangfeng Chen, Huanzhang Lu
National Key Laboratory of Automatic Target Recognition (ATR), School of Electronic Science and Engineering, National University of Defense Technology, Changsha, Hunan, 410073 - China
[e-mail: huiwuluo@nudt.edu.cn]
*Corresponding author: Huiwu Luo

Received October 3, 2016; revised December 13, 2016; accepted December 27, 2016; published March 31, 2017

Abstract

Bag of visual words is a popular model in human action recognition, but usually suffers from loss of spatial and temporal configuration information of local features, and large quantization error in its feature coding procedure. In this paper, to overcome the two deficiencies, we combine sparse coding with spatio-temporal pyramid for human action recognition, and regard this method as the baseline. More importantly, which is also the focus of this paper, we find that there is a hierarchical structure in feature vector constructed by the baseline method. To exploit the hierarchical structure information for better recognition accuracy, we propose a tree regularized classifier to convey the hierarchical structure information. The main contributions of this paper can be summarized as: first, we introduce a tree regularized classifier to encode the hierarchical structure information in feature vector for human action recognition. Second, we present an optimization algorithm to learn the parameters of the proposed classifier. Third, the performance of the proposed classifier is evaluated on YouTube, Hollywood2, and UCF50 datasets, the experimental results show that the proposed tree regularized classifier obtains better performance than SVM and other popular classifiers, and achieves promising results on the three datasets.

Keywords: Bag of visual words, feature coding, SVM, structure sparsity, spatio-temporal pyramid

https://doi.org/10.3837/tiis.2017.03.020 **ISSN : 1976-7277**

1. Introduction

Human action recognition has been an important research area in computer vision for several decades, which has been widely used in video retrieval, human-computer interaction, etc. In recent years, bag of visual words (BOVW) has been verified as an both effective and efficient model for human action recognition. Generally, the application of traditional BOVW for human action recognition has three steps: feature extraction, feature coding, and pooling of all coding coefficients over the whole video to form the final feature vector [1]. However, there are two well-known drawbacks for the traditional BOVW model used for human action recognition, and many approaches have been proposed to cope with them.

On the one hand, the traditional BOVW uses hard assignment (i.e. assign each feature to a visual word) in the second step, resulting in un-negligible quantization error. Thus a variety of soft assignment methods have been introduced to reduce the quantization error by assigning each feature to several visual words [1], such as sparse coding [2], kernel codebook [3], locality-constrained linear encoding [4], and Fisher kernel [5], etc. Among these methods, sparse coding has been validated as an effective feature coding method in human action recognition [1], which is often followed by max pooling operation [2].

On the other hand, for the traditional BOVW model, the feature vector of a video is generated by pooling of all coding coefficients over the whole video, failing to capture the informative spatial and temporal configuration information of local features, leading to inferior performance. Many approaches have been proposed to overcome this problem, and most of these approaches can be divided into two categories: context codebook construction [6-8] and spatio-temporal pyramid [9-11]. The context codebook construction approach takes into account the spatial and temporal neighbors of local features and utilize them to construct a context codebook, so the feature vector calculated on the context codebook has captured the spatial and temporal contextual information of local features [6-8]. The spatio-temporal pyramid is derived from the spatial pyramid used in image classification [12]. The basic idea of spatio-temporal pyramid is to divide the video into several regions, then take the concatenation of the feature vectors of all regions as the final feature vector of a video. Compared with context codebook construction approach, the spatio-temporal pyramid is easy to implement, which has been widely applied in human action recognition [9-11].

To overcome the two deficiencies of traditional BOVW and make use of the latest achievements in human action recognition, we combine sparse coding [2] with spatio-temporal pyramid [9] for human action recognition, and regard this method as the baseline in this paper.

Generally speaking, the feature vector generated by the baseline method should be classified by support vector machine (SVM), since the SVM has been the most common classifier in human action recognition [1, 6-11]. However, the SVM uses L2 regularizer in its learning process, and the L2 regularizer penalizes each of the variables in feature vector separately, which can only convey trivial prior information about the feature vector [13]. In some cases, the variables in feature vector have structure relationships [14, 15]. For example, if a variable in feature vector is unrelated to a given classification task, some other variables will be unrelated, either, which is known as hierarchical structure [15]. The structure information of feature vector can be regarded as prior information for the classifier in the Bayesian framework [14, 15], thus a variety of structured regularizers have been proposed to

convey the prior information to the classifier, such as group regularizer [14], tree regularizer [15], etc. More importantly, the structure information embedded in classifier cannot only express prior information about the intended sparsity patterns, but also boost the classification accuracy [13], which has been successfully applied in text categorization [16, 17], display advertising [18], etc.

In this paper, we find that the feature vector generated by the baseline method has a hierarchical structure [15]. Since the L2 regularizer in SVM cannot convey the hierarchical structure prior information. To exploit the hierarchical structure information for better recognition accuracy, we proposed a tree regularized classifier for human action recognition by substituting L2 regularizer for tree regularizer in SVM.

However, optimizing of tree regularized classifier is challenging, since both of tree regularizer and hinge loss function in the proposed tree regularized classifier are non-differentiable. Many efforts [19-24] have studied the optimization problem of structure regularized penalty, which can be split into three categories: subgradient method, proximal method [20] and ADMM [24]. The subgradient method can be applied to arbitrary structure regularized penalty, but it is very slow to converge. The proximal method has more faster convergence rate, but it can only solve limited kinds of structure regularized penalties [20, 22]. The basic idea of ADMM is to partition the optimization problem into sub-problems, which can be applied to large scale optimization problem [24]. Besides, ADMM is more universal and can hand a variety of structure regularized penalties, especially for the structure regularizer with overlapping groups [19, 23]. All in all, although many methods have been proposed to solve the structure regularized penalty problems, they mainly addressed the optimization problems of structure regularized penalty combined with the differentiable loss function, such as logistic loss function. But the hinge loss function in the proposed tree regularized classifier is non-differentiable, in this paper, we show how to deal with the optimization problem of the tree regularized classifier using ADMM method [24].

We employ YouTube [26], Hollywood2 [25], and UCF50 [32] datasets to evaluate the proposed tree regularized classifier, and experimental results reveal that the tree regularized classifier gets better performance than SVM and other popular classifiers, and achieves promising recognition accuracy on the datasets.

The main contributions of this paper can be summarized as: first, we introduce a tree regularized classifier to encode the hierarchical structure information in feature vector for human action recognition. Second, we present an optimization algorithm to learn the parameters of the proposed classifier, and the performance of the proposed classifier is evaluated on three public available datasets.

The rest of this paper is organized as follows: we introduce the baseline method in Section 2, including the generation of the feature vector and the learning procedure of the frequently used SVM. In Section 3, we first illustrate the hierarchical structure in feature vector generated by the baseline method, then we adopt the tree regularizer to encode the hierarchical structure information of feature vector, and solve the optimization problem using ADMM method. The proposed tree regularized classifier is evaluated in Section 4, and the conclusion is given in Section 5.

2. Baseline Method

2.1 Local Features

We extract dense trajectories [9] from video, and character each trajectory with the concatenation of four types of descriptors, i.e., trajectory shape, histogram of optical flow (HOF), histogram of oriented gradients (HOG) and motion boundary histogram (MBH). We cluster the set of descriptors extracted from videos to construct a codebook. Then we code each local feature using sparse coding [2], and obtain the coding coefficient for each local feature.

2.2 Spatio-Temporal Pyramid

After the coding coefficient of each local feature has been calculated by sparse coding [2], we employ spatio-temporal pyramid to capture the spatial and temporal configuration information of local features.

Following the setup in [9], we use six types of spatio-temporal grids, which are denoted by $1\times1\times1$, $3\times1\times1$, $2\times2\times1$, $1\times1\times2$, $3\times1\times2$, $2\times2\times2$ shown in **Fig. 1**, where the first dimension corresponds to horizontal subdivision, the second dimension corresponds to vertical subdivision and the third dimension corresponds to temporal subdivision. Thus, the 24 regions have been constructed by the six grids, denoted by $r_1,\ldots, r_{24}$ shown in **Fig. 1** (the obscured r_{23} is not shown).

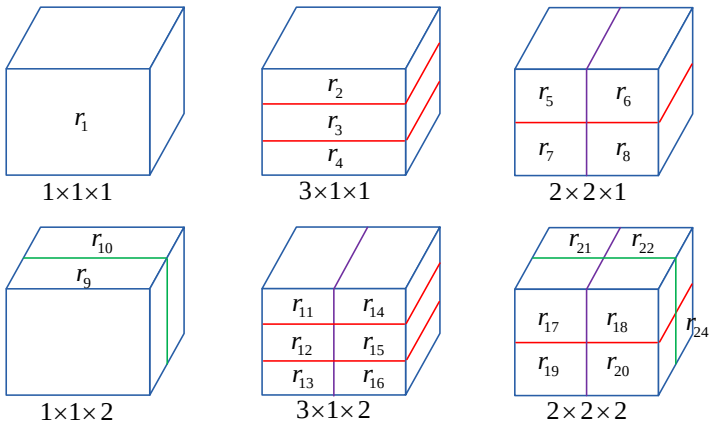

Fig. 1. The six types of spatio-temporal grids

After the spatio-temporal pyramid has been constructed, we calculate the feature vector of each region by max pooling all coding coefficients in the region, and the final feature vector of a video is formed by concatenating the feature vectors of all 24 regions. In this paper, we take the feature vector generated in this way (i.e. sparse coding combined with spatio-temporal pyramid) as the baseline method.

2.3 SVM

Generally, the above feature vector (i.e. the feature vector generated by the baseline method) is usually classified by SVM, since SVM has been the most common classifier in human action

recognition [1, 6-11]. We assume the feature vector of a video is denoted by $\boldsymbol{x}$, and $\boldsymbol{x} \in \mathrm{R}^{p}$. The learning problem of SVM $\boldsymbol{x}^{T}\boldsymbol{\omega}^{*}+b^{*}$ can be written as:

$$\left(\boldsymbol{\omega}^{*}, b^{*}\right)=\arg\min_{\boldsymbol{\omega}, b}\sum_{i=1}^{n}\left[1-y_{i}\left(\boldsymbol{\omega}^{T}\boldsymbol{x}_{i}+b\right)\right]_{+}+\lambda\Omega(\boldsymbol{\omega}) \tag{1}$$

where $[a]_{+}=\max(a,0)$ is the hinge loss function, y_i is the class label for the ith video, $\boldsymbol{\omega}$ is the weight vector, and b is a bias. $\Omega(\boldsymbol{\omega})$ is the L2 regularizer, and $\Omega(\boldsymbol{\omega})=\|\boldsymbol{\omega}\|_{2}^{2}=\sum_{j=1}^{p}\omega^{2}(j)$ in SVM. $\lambda>0$ is the regularization parameter.

As we can see, the L2 regularizer penalizes each of the variables separately, which can only convey trivial prior information about the feature vector. In some cases, the variables in feature vector have informative relationship, and the relationship can be regarded as prior information for the classifier, which can boost the classification accuracy by encoding it into classifier through corresponding structure regularizer. More importantly, the structure regularizers have been applied in many areas [16-18].

To exploit the structure regularizer for better classification accuracy, we must find the relationship between the variables in feature vector, i.e., the structure of feature vector.

3. The Tree Regularized Classifier

3.1 The Hierarchical Structure in Feature Vector

To illustrate the hierarchical structure in feature vector generated by the baseline method, we review the max pooling procedure and take an example to illustrate the hierarchical structure, since the hierarchical structure of feature vector introduced in this paper is derived from the max pooling operation.

We assume the codebook size is 3, and M local features are extracted from a video. We can obtain M coding coefficients for the M local features after sparse coding [2]. Let these coding coefficients be denoted by $L=\{\boldsymbol{l}_{1},\ldots\boldsymbol{l}_{\mathrm{M}}\}$, and $\boldsymbol{l}_{i}\in R^{3}$. We assume the feature vector of the video is denoted by $\boldsymbol{x}$, which can be calculated by concatenating the feature vectors of all 24 regions, thus $\boldsymbol{x}\in R^{72}$. Meanwhile, $(x(1),x(2),x(3))$ is the feature vector of the region r_1, and $(x(4),x(5),x(6))$ is the feature vector of the region r_2, etc. They are calculated by max pooling of all of the coding coefficients in corresponding regions, which can be written as:

$$\begin{aligned}
&(x(1),x(2),x(3))=\max\left\{\boldsymbol{l}_{\tau}\right\}_{\boldsymbol{l}_{\tau}\subseteq r_{1}},\\
&(x(4),x(5),x(6))=\max\left\{\boldsymbol{l}_{\tau}\right\}_{\boldsymbol{l}_{\tau}\subseteq r_{2}},\\
&\qquad\ldots\\
&(x(70),x(71),x(72))=\max\left\{\boldsymbol{l}_{\tau}\right\}_{\boldsymbol{l}_{\tau}\subseteq r_{24}}.
\end{aligned} \tag{2}$$

where $\boldsymbol{l}_{\tau}\subseteq r_{\eta}$ denotes all coding coefficients belonging to the region r_{η}. Besides, the first dimension of the feature vector for each region (e.g., $x(1)$, $x(4)$, $x(7)$,…, $x(70)$) is the coding coefficient coded on the first visual word, and the second dimension of each region (e.g., $x(2)$, $x(5)$, $x(8)$,…, $x(71)$) is the coding coefficient coded on the second visual word, etc.

Moreover, from the spatio-temporal pyramid shown in **Fig. 1**, we can observe that the coding coefficients in the region r_1 are the union of all the coding coefficients in the regions $r_2 \sim r_{24}$:

$$\{\boldsymbol{l}_\tau\}_{\boldsymbol{l}_\tau \subseteq r_1} = \{\boldsymbol{l}_\tau\}_{\boldsymbol{l}_\tau \subseteq r_2} \bigcup \{\boldsymbol{l}_\tau\}_{\boldsymbol{l}_\tau \subseteq r_3} \cdots \bigcup \{\boldsymbol{l}_\tau\}_{\boldsymbol{l}_\tau \subseteq r_{24}} \tag{3}$$

Therefore, we can obtain the relationship of the variables in feature vector $\boldsymbol{x}$:

$$\begin{aligned} x(1) &= \max(x(4), x(7), x(10), \ldots x(70)), \\ x(2) &= \max(x(5), x(8), x(11), \ldots x(71)), \\ x(3) &= \max(x(6), x(9), x(12), \ldots x(72)). \end{aligned} \tag{4}$$

The above relationship about the feature vector $\boldsymbol{X}$ can be known as hierarchical structure [15] described by three trees shown in **Fig. 2**, which are composed of 72 nodes, and the numbers in the nodes are the indices for $\boldsymbol{X}$. Concretely, the nodes belonging to the same tree correspond to the same visual word in codebook, and the value of root node is the maximum of all its children in each tree. Furthermore, the nodes painted with the same background color are from the same region, e.g., the root nodes (i.e., 1, 2, 3) are the feature vector generated by the region r_1.

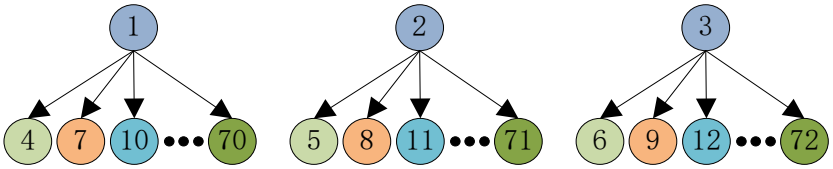

Fig. 2. The hierarchical structure of feature vector

As we can see from the equation (4), if the value of any leaf node in **Fig. 2** is non-zero, its parent node will be non-zero. On the contrary, if the value of any parent node is zero, all its leaf nodes should be zero, too. As illustrated in [2], the max pooling operator can be seen as the selection of the most salient feature in a region, e.g., the root node can be seen as the most salient feature in the whole video (i.e. the region r_1). Therefore, we can conclude that if the most salient feature in a coarse region is not discriminative, all features corresponding to the same visual word in all finer regions which partitioned by this coarse region are not discriminative, either. For example, $x(1)$ is the most salient feature in r_1 (i.e. the whole video), and corresponds to the first visual word in codebook, thus, if $x(1)$ is not discriminative, all features corresponds to the first visual word in all finer regions which partitioned by r_1 (i.e. $r_2 \sim r_{24}$) are not discriminative (i.e. $x(4), x(7), x(10), \ldots, x(70)$), either.

Now we have found the hierarchical structure of feature vector, since the L2 regularizer in SVM cannot express the hierarchical structure, for better classification accuracy, we employ tree regularizer to express it, and propose the tree regularized classifier by replacing the L2 regularizer with tree regularized in SVM. We will show how to apply the tree regularizer to encode the hierarchical structure illustrated above.

3.2 Tree Regularizer

Formally, we assume the size of the codebook is S, as clarified above, the hierarchical structure of feature vector $\boldsymbol{x} \in \mathrm{R}^p$ can be described by S trees. We assume these trees are indexed by k, k =1,…, S. The number of nodes in the kth tree is m_k (all m_k are equal to 24 in this paper), the nodes in the kth tree are indexed by j, j =1,…, m_k. The tree regularizer [15] can be written as:

$$\Omega(\boldsymbol{\omega}) = \sum_{k=1}^{S} \sum_{j=1}^{\mathrm{m}_k} \varepsilon_j^k \left\| \boldsymbol{\omega}_{G_j^k} \right\|_2 \tag{5}$$

where G_j^k is a non-empty subset of $\{1,..., p\}$, indexing the jth node and all its children in the kth tree. $\boldsymbol{\omega}_{G_j^k} = \left[\boldsymbol{\omega}(i)\right]_{i \in G_j^k}$ is a sub-vector of $\boldsymbol{\omega}$. $\varepsilon_j^k > 0$ is a regularization parameter for the group (k, j). Similar to [15], we define $\varepsilon_j^k = \sqrt{d_j^k}$, where d_j^k is the number of elements in the (k, j) group.

For example, the tree regularizer for the example shown in **Fig. 2** is:

$$\begin{aligned} \Omega(\boldsymbol{\omega}) &= \sqrt{21}\left\|(\boldsymbol{\omega}(1), \boldsymbol{\omega}(4),..., \boldsymbol{\omega}(70))\right\|_2 + \left|\boldsymbol{\omega}(4)\right| +,..., + \left|\boldsymbol{\omega}(70)\right| \\ &+ \sqrt{21}\left\|(\boldsymbol{\omega}(2), \boldsymbol{\omega}(5),..., \boldsymbol{\omega}(71))\right\|_2 + \left|\boldsymbol{\omega}(5)\right| +,..., + \left|\boldsymbol{\omega}(71)\right| \\ &+ \sqrt{21}\left\|(\boldsymbol{\omega}(3), \boldsymbol{\omega}(6),..., \boldsymbol{\omega}(72))\right\|_2 + \left|\boldsymbol{\omega}(6)\right| +,..., + \left|\boldsymbol{\omega}(72)\right| \end{aligned} \tag{6}$$

where $\left\|\boldsymbol{\omega}(i)\right\|_2 = \left|\boldsymbol{\omega}(i)\right|$ is the absolute value of $\boldsymbol{\omega}(i)$. The effect of the tree regularizer [15] is: a feature can be selected only if its parent is selected in a tree, in other words, the weight of a feature is non-zero only if the weight of its parent is non-zero [15]. For example, $\boldsymbol{\omega}(4)$ will be non-zero only if $\boldsymbol{\omega}(1)$ is non-zero. Similarly, if $\boldsymbol{\omega}(1)$ is zero, all its children $\boldsymbol{x}(4), \boldsymbol{x}(7), \boldsymbol{x}(10),..., \boldsymbol{x}(70)$ will be induced to zero.

3.3 Optimization of The Tree Regularized Classifier

We now turn our attention to the optimization problem of the tree regularized classifier. We substitute tree regularizer (5) for L2-norm regularizer (1) in SVM, and the optimization problem of tree regularized classifier can be written as:

$$\left(\boldsymbol{\omega}^*, b^*\right) = \min_{\boldsymbol{\omega}, b} \sum_{i=1}^{n} \left[1 - y_i\left(\boldsymbol{\omega}^T \boldsymbol{x}_i + b\right)\right]_+ + \sum_{k=1}^{S} \sum_{j=1}^{\mathrm{m}_k} \lambda_j^k \left\| \boldsymbol{\omega}_{G_j^k} \right\|_2 \tag{7}$$

where $\lambda_j^k = \lambda \varepsilon_j^k$. The main challenges of the optimization problem (7) are that not only both of the hinge loss function and the tree regularizer are non-differentiable, but also the groups in tree regularizer are overlapping [15,16].

We first introduce an auxiliary variable $\boldsymbol{v}$ to transform the optimization problem (7) with overlapping groups into the following equivalent problem:

$$\begin{aligned} &\min_{\boldsymbol{\omega}, b} \sum_{i=1}^{n} \left[1 - y_i\left(\boldsymbol{\omega}^T \boldsymbol{x}_i + b\right)\right]_+ + \sum_{k=1}^{S} \sum_{j=1}^{\mathrm{m}_k} \lambda_j^k \left\| \boldsymbol{v}_{D_j^k} \right\|_2 \\ &s.t. \quad \boldsymbol{v} = \boldsymbol{\varphi} \end{aligned} \tag{8}$$

where $\boldsymbol{\varphi} = \left[\boldsymbol{\omega}_{G_1^1}, ..., \boldsymbol{\omega}_{G_{m_1}^1}, \boldsymbol{\omega}_{G_1^2}, ..., \boldsymbol{\omega}_{G_{m_S}^S} \right]^T \in \mathrm{R}^t$, $\boldsymbol{v} = \left[\boldsymbol{v}_{D_1^1}, ..., \boldsymbol{v}_{D_{m_1}^1}, \boldsymbol{v}_{D_1^2}, ..., \boldsymbol{v}_{D_{m_S}^S} \right]^T$, and the cardinality of D_j^k is equal to the cardinality of G_j^k, but indies $\boldsymbol{v}$ with increasing number. We can see that the groups in regularizer term are no longer overlapping.

Inspired from [19], we introduce a sparse matrix $\boldsymbol{M} \in \mathrm{R}^{t \times p}$, only one element is 1 in each row of $\boldsymbol{M}$ (others are 0), and the element of 1 corresponds to the type of visual word of $\varphi(i)$, thus $\boldsymbol{\varphi} = \boldsymbol{M}\boldsymbol{\omega}$, and (8) can be equivalently written as:

$$\begin{aligned} &\min_{\boldsymbol{\omega},b} \sum_{i=1}^{n} \left[1 - y_i \left(\boldsymbol{\omega}^T \boldsymbol{x}_i + b \right) \right]_+ + \sum_{k=1}^{S} \sum_{j=1}^{m_k} \lambda_j^k \left\| \boldsymbol{v}_{D_j^k} \right\|_2 \\ &s.t. \quad \boldsymbol{v} = \boldsymbol{M}\boldsymbol{\omega} \end{aligned} \tag{9}$$

The optimization problem (9) can be solved by iterating the following parameters alternately.

$$\begin{aligned} &\left(\boldsymbol{\omega}^{(t+1)}, b^{(t+1)} \right) = \arg\min_{\boldsymbol{\omega},b} \sum_{i=1}^{n} \left[1 - y_i \left(\left(\boldsymbol{\omega}^{(t)} \right)^T \boldsymbol{x}_i + b^{(t)} \right) \right]_+ + \frac{\rho}{2} \left\| \boldsymbol{M}\boldsymbol{\omega}^{(t)} - \boldsymbol{v}^{(t)} + \boldsymbol{u}^{(t)} \right\|_2^2 \\ &\boldsymbol{v}^{(t+1)} = \arg\min_{\boldsymbol{v}} \sum_{k=1}^{S} \sum_{j=1}^{m_k} \lambda_j^k \left\| \left(\boldsymbol{v}^{(t)} \right)_{D_j^k} \right\|_2 + \frac{\rho}{2} \left\| \boldsymbol{M}\boldsymbol{\omega}^{(t+1)} - \boldsymbol{v}^{(t)} + \boldsymbol{u}^{(t)} \right\|_2^2 \\ &\boldsymbol{u}^{(t+1)} = \boldsymbol{u}^{(t)} + \boldsymbol{M}\boldsymbol{\omega}^{(t+1)} - \boldsymbol{v}^{(t+1)} \end{aligned} \tag{10}$$

where $\boldsymbol{u}$ is the scaled dual variable [24], and $\rho > 0$ is a penalty parameter. We show the details for the update of $(\boldsymbol{\omega}, b)$ and $\boldsymbol{v}$ in the following.

For the update of $(\boldsymbol{\omega}, b)$ in (10), we can form an equivalently problem by introducing an auxiliary variable $\boldsymbol{\xi} \in R^n$.

$$\begin{aligned} &\left(\boldsymbol{\omega}^{(t+1)}, b^{(t+1)} \right) = \arg\min_{\boldsymbol{\omega},b} \sum_{i=1}^{n} \xi_i + \frac{\rho}{2} \left\| \boldsymbol{M}\boldsymbol{\omega}^{(t)} - \boldsymbol{v}^{(t)} + \boldsymbol{u}^{(t)} \right\|_2^2 \\ &s.t. \quad y_i \left(\left(\boldsymbol{\omega}^{(t)} \right)^T \boldsymbol{x}_i + b^{(t)} \right) \geq 1 - \xi_i, \\ &\qquad \xi_i \geq 0, \qquad\qquad i = 1, ..., n. \end{aligned} \tag{11}$$

The dual problem of (11) can be written as:

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} -\frac{1}{2\rho} \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i^T \left(\boldsymbol{M}^T \boldsymbol{M} \right)^{-1} \sum_{j=1}^{n} \alpha_j y_j \boldsymbol{x}_j - \sum_{i=1}^{n} \alpha_i \left[\left(\boldsymbol{v} - \boldsymbol{u} \right)^T \boldsymbol{M} \left(\boldsymbol{M}^T \boldsymbol{M} \right)^{-1} y_i \boldsymbol{x}_i - 1 \right] \\ &s.t. \ 0 \leq \alpha_i \leq 1 \ , \ i = 1, 2, ..., n. \\ &\sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned} \tag{12}$$

where $\boldsymbol{\alpha} \in R^n$ is the Lagrange multiplier, and the matrix $\boldsymbol{M}^T\boldsymbol{M}$ is diagonal and invertible, thus it is convenient to calculate the inverse matrix of $\boldsymbol{M}^T\boldsymbol{M}$. Besides, we observe that the optimization problem (12) is a quadratic programming problem, thus we can solve it via sequential minimization optimization (SMO) [27], and the update of $\boldsymbol{\omega}$ can be written as:

$$\boldsymbol{\omega}^{(t+1)} = \frac{1}{\rho}\left(\boldsymbol{M}^T\boldsymbol{M}\right)^{-1}\left[\sum_{i=1}^{n}\alpha_i y_i \boldsymbol{x}_i + \rho\boldsymbol{M}^T\left(\boldsymbol{v}^{(t)} - \boldsymbol{u}^{(t)}\right)\right] \tag{13}$$

Whereas the update for b can be written as:

$$b^{(t+1)} = \frac{\sum_{i\in\mathbf{S}^*}\left(y_i - \left(\boldsymbol{\omega}^{(t+1)}\right)^T\boldsymbol{x}\right)}{\left|\mathbf{S}^*\right|} \tag{14}$$

where $\mathbf{S}^* = \left\{i \mid 0 < \alpha_i^* < 1\right\}$, and $\left|\mathbf{S}^*\right|$ is the cardinality of $\mathbf{S}^*$.

For the update of $\boldsymbol{v}$ in (10), we can equivalently reformulate it as:

$$\boldsymbol{v}^{(t+1)} = \arg\min_{\boldsymbol{v}}\sum_{k=1}^{S}\sum_{j=1}^{\mathrm{m}_k}\left(\lambda_j^k\left\|\left(\boldsymbol{v}^{(t)}\right)_{D_j^k}\right\|_2 + \frac{\rho}{2}\left\|\left(\boldsymbol{M}\boldsymbol{\omega}^{(t+1)}\right)_{D_j^k} - \left(\boldsymbol{v}^{(t)}\right)_{D_j^k} + \left(\boldsymbol{u}^{(t)}\right)_{D_j^k}\right\|_2^2\right) \tag{15}$$

We solve it using block soft thresholding operator [20]:

$$\left(\boldsymbol{v}^{(t+1)}\right)_{D_j^k} = \left(1 - \frac{\lambda_j^k}{\rho\left\|\boldsymbol{\sigma}_{D_j^k}\right\|_2}\right)_+ \boldsymbol{\sigma}_{D_j^k},\ k = 1,\ldots s;\ j = 1,\ldots,\mathrm{m}_k. \tag{16}$$

where $\boldsymbol{\sigma} = M\boldsymbol{\omega}^{(t+1)} + \boldsymbol{u}^{(t)}$, $\boldsymbol{\sigma}_{D_j^k} = \left(\boldsymbol{\sigma}_i\right)_{i\in D_j^k}$. $\left(a\right)_+$ is the hinge loss function. When $\left\|\boldsymbol{\sigma}_{D_j^k}\right\|_2 = 0$, $\left(\boldsymbol{v}^{(t)}\right)_{D_j^k} = \mathbf{0}$.

For the proof property of the proposed optimization algorithm, we can derive it from the convergence theory of ADMM [24] easily.

4. Experiments

4.1 Experimental Setup

We employ three popular human action datasets to evaluate the proposed baseline method and the tree regularized classifier, i.e. YouTube [26], Hollywood2 [25], and UCF50 [32].

The purposes of the experiments are to validate whether the baseline method illustrated in Section 2 can improve the performance of the traditional BOVW and the Fisher kernel method [5], whether the proposed tree regularized classifier can boost the recognition accuracy further, and whether the proposed tree regularized classifier outperforms other classifiers.

Therefore, we conduct three parts of experiments: we first evaluate the performance of baseline method and the proposed tree regularized classifier on the three datasets, which is shown in Section 4.2, Section 4.3, and Section 4.4, respectively. Then we compare the proposed tree regularized classifier with other classifiers, which is presented in Section 4.5. Finally, we compare the recognition accuracy obtained by the proposed method with the state of the art methods, which is shown in Section 4.6.

In all of the experiments, as mentioned in Section 2, we extract dense trajectories [9] from video, and character each trajectory with the concatenation of four types of descriptors, i.e., trajectory shape, HOF, HOG, and MBH. We construct the codebook using K-means clustering

method, and the parameters for sparse coding in the baseline method are the same as in [1]. With regard to the fisher kernel method, we generate the fisher vector as the same fashion illustrated in [5]. In regard to the implementation of SVM, we use linear SVM implemented by LIBSVM [28]. Specifically, we train the classifier using the one-vs.-rest strategy for the multiclass classification.

Regarding the optimization parameters of the proposed tree regularized classifier, the best regularization parameter λ and the best penalty parameters ρ are selected from $\{2^{-14}, 2^{-13}, ..., 2^{8}\}$ by cross validation.

4.2 Evaluation of Recognition Accuracy on YouTube Dataset

The YouTube [26] dataset includes 11 actions, and has 1168 video clips in total. Recognizing the human actions of the video clips in YouTube dataset is very challenging, since these video clips are recorded under cluttered background. The example frames sampled from YouTube dataset are shown in **Fig. 3**.

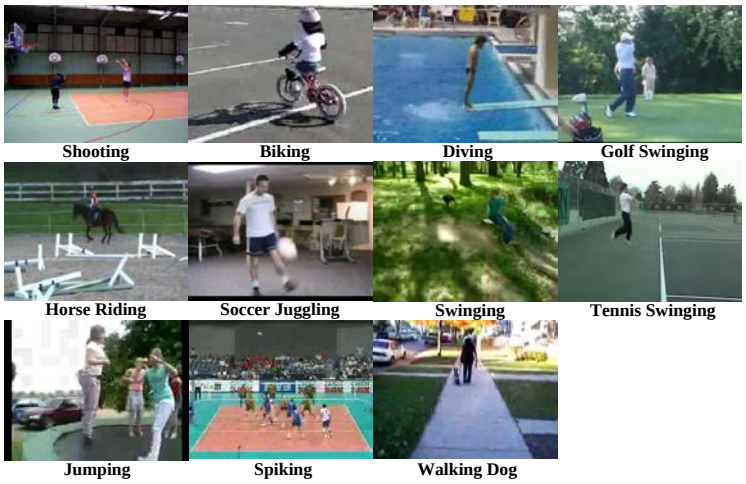

Fig. 3. The example frames sampled from YouTube dataset

We generate the feature vector using three different methods (i.e., the traditional BOVW method, the fisher kernel method, and the baseline method illustrated in Section 2), and conduct the experiments with different codebook sizes, i.e., for the BOVW and baseline method, we evaluate the performance with the size of 1k, 2k, 4k, 6k, 8k, 10k, respectively. Whereas for the fisher kernel method, the number of Gaussians K is set to 256, 512, 1024, respectively, we then compare the performance of the proposed tree regularized with SVM. We use leave one out cross validation to train the classifier, i.e., 24 pre-defined groups are used for training and the rest one group is used for testing, as in [9], and the average accuracy over all classes is presented, the experimental results are shown in **Table 1**.

Table 1. Evaluation of the recognition accuracy on YouTube dataset

Feature vector generation method	Classifier	Codebook size					
		1k	2k	4k	6k	8k	10k
Traditional BOVW	SVM	74.5	76.9	78.8	79.6	78.1	76.5
Fisher kernel	SVM	83.6 (K=256)		85.4 (K=512)		84.9 (K=1024)	
Baseline	SVM	78.1	81.3	85.3	86.1	86.4	87
Baseline	Tree regularized classifier	78.3	81.7	86.2	87.1	87.5	88.3

As we can see from **Table 1**, basically, the recognition accuracy show continuous improvement with the increasing of the codebook size except the traditional BOVW generation method. Besides, the baseline method outperforms both of fisher kernel method and traditonal BOVW significantly, e.g., when the codebook size is set to 6000, the recognition accuracy achieved by traditional BOVW is 79.6%, whereas the baseline method obtains 86.1%, which has improved by 6.5%, and the fisher kernel method achieves 85.4% on the dataset, which is inferior to the performance of baseline method.The reason for the improvement is that the baseline method employes sparse coding to reduce the quantization error and the spatio-temporal pyramid to capture the informative spatio-temporal configuration information of local features. What is more, the tree regularized classifier has improved the performance of SVM, e.g., when the codebook size is set to 6000, the tree regularized classifier has achieved 87.1%, which has improved the performance of SVM by 1%.

We then evaluate performance of the tree regularized classifier with different iterations on YouTube dataset, the codebook size is set to 6000, the results are shown in **Table 2**.

Table 2. Evaluation of the recognition accuracy with different iterations on YouTube dataset

Iterations	100	200	300	400	500	600	700
Recognition accuracy	35.2	43.2	61.2	71.8	87.1	87.2	87.2

From **Table 2**, we notice that the recognition accuracy obtained by tree regularized classifier increases with the increasing of the iterations, when the iterations are higher than 500, the growth of the recognition accuracy becomes slowly. Therefore, to get a compromise between the computational time and the recognition accuracy, we implement 500 iterations for the train of the proposed tree regularized classifier in all the experiments, which empirically shows satisfactory results.

4.3 Evaluation of Recognition Accuracy on Hollywood2 Dataset

The video clips in Hollywood2 [25] dataset are taken from Hollywood movies, which can be divided into 12 action classes: driving car, eating, getting out of car, hand shaking, hugging person, answering phone, standing up, sitting up, sitting down, running, kissing, and fighting person. Besides, there are 1707 video clips in Hollywood2 dataset, which can be split into the training set (823 video clips) and the test set (884 video clips). The example frames sampled from Hollywood2 dataset are shown in **Fig.4**.

Fig. 4. The example frames sampled from Hollywood2 dataset

With regard to the experiments evaluated on Hollywood2 dataset, we report the mean average precision (mAP) [25] over all action classes, the experimental results are shown in **Table 3**.

Table 3. Evaluation of the recognition accuracy on Hollywood2 dataset

Feature vector generation method	Classifier	Codebook size					
		1k	2k	4k	6k	8k	10k
Traditional BOVW	SVM	48.1	50.8	53.2	54.1	54.9	54.6
Fisher kernel	SVM	58.5 (K=256)		60.4 (K=512)		59.7 (K=1024)	
Baseline	SVM	55.2	57.3	61.5	62.8	63.9	64.6
Baseline	Tree regularized classifier	55.3	57.6	62.3	63.7	65	65.6

From **Table 3**, we notice that the baseline method has achieved better performance than traditional BOVW and fisher kernel, and large-sized codebook always results in better performance. The reason for the phenomenon can be explained as: the feature used in our experiments is sampled from video densely, the densely sampled features can describe more informative information and then need a large-sized codebook to be expressed. Also, we can see that the tree regularized classifier has improved the performance of SVM further. For example, when the codebook size is set to 6000, the traditional BOVW has obtained 54.1%, and the baseline has achieved 62.8%, whereas the tree regularized classifier has achieved 63.7%, which is 0.9% higher than that of SVM.

4.4 Evaluation of Recognition Accuracy on UCF50 Dataset

The UCF50 [32] dataset consists of realistic video clips collected from YouTube, which contains 50 actions, such as baseball pitch, bench press, clean and jerk, drumming, fencing, playing guitar, pizza tossing, etc. Similarly, the video clips belonging to the same action are split into 25 groups for the UCF50 dataset. Some example frames sampled from UCF50 dataset are shown in **Fig. 5**.

Fig. 5. Some example frames sampled from UCF50 dataset

In the experiments, we adopt leave one out cross validation for the pre-defined 25 groups, and the average accuracy over all classes is reported, the experimental results are presented in **Table 4**.

Table 4. Evaluation of the recognition accuracy on UCF50 dataset

Feature vector generation method	Classifier	Codebook size					
		1k	2k	4k	6k	8k	10k
Traditional BOVW	SVM	69.8	72.5	74.3	75.8	76.4	77.5
Fisher kernel	SVM	84.2 (K=256)		85.7 (K=512)		85.1 (K=1024)	
Baseline	SVM	77.5	83.7	85.3	86.9	87.2	88
Baseline	Tree regularized classifier	77.6	84.2	86.1	87.9	88.1	89.1

From **Table 4**, as before, the baseline method has achieved better performance than traditional BOVW and fisher kernel method, and the tree regularized classifier outperforms SVM. When the size of codebook is set to 1000, 2000, 4000, 6000, 8000, 10000, respectively, the recognition accuracy achieved by tree regularized classifier is 0.1%, 0.5%, 0.8%, 1%, 0.9%, 1.1% higher than SVM, respectively.

In the above three datasets, we can observe that the baseline method obtains better performance than fisher kernel method, the reason for this performance is that the baseline method has captured informative spatial and temporal information of local features.

Besides, The reasons for the superior performance of tree regularized classifier can be summarized as: on the one hand, as we can see from the feature vector generation process of the baseline method illustrated in Section 2, the feature vector for a video is generated by concatenating the feature vectors of all 24 regions, that is, the length of the feature vector is 24 times of the size of codebook, e.g., if the size of codebook is 2000, the length of the feature vector generated by baseline method is 48000 (24*2000). Obviously, for this type of high dimensional classification problem, the weight vector of the classifier should be sparse, more importantly, the weight vector has a hierarchical sparsity pattern illustrated in Section 3.1.

Unfortunately, the SVM cannot result in sparse weight vector directly, and does not take advantage of the inherent structure either. However, the tree regularized classifier takes into account the inherent hierarchical structure of the feature vector, and outperforms SVM undoubtedly. On the other hand, as illustrated in Section 1, the sparsity pattern of the weight vector can be seen as prior information for the classifier under the Bayesian framework [14, 15], and the improvement classification performance is benefited from the inherent prior information.

4.5 Comparison with Other Classifiers

We now compare the tree regularized classifier with other popular classifiers, including L1-norm regularized logistic regression based classifier (L1_LRC), L2-norm regularized logistic regression based classifier (L2_LRC), sparse representation based classifier (SRC) [39], collaborative representation based classifier (CRC) [40], random forest (RF), L1-norm regularized SVM (L1_SVM) [41] and SVM. The parameters in these classifiers are determined by cross validation. For the implementation of these classifiers, both of L1_LRC and L2_LRC are implemented by SPAMS package [38], the L1 regularized minimization problem in SRC [39] is solved by [43], and CRC is implemented by [40], RF is implemented by the function in Matlab, L1_SVM is implemented by LIBLINEAR [41]. We evaluate the recognition accuracy on the three datasets, and record the running time of implementing a classifier on YouTube dataset. The computer configuration in the experiments is Intel I5 3450 and 16GB memory. We implement the experiments using Matlab2014. The comparison results are shown in **Table 5.**

Table 5. Comparison with other classifiers

Method	**Classification accuracy on datasets**			**Running time on YouTube**
	YouTube	Hollywood2	UCF50	
L1_LRC	78.2	56.1	77.1	0.48s
L2_LRC	83.9	62.3	81.9	1.12s
SRC	81.1	60.9	82.3	108.31s
CRC	80.2	61.3	81.6	0.19s
RF	84.5	63.7	82.4	20.81s
L1_SVM	79.7	58.5	79.6	0.002s
SVM	87	64.6	88	2.24s
Our method	88.3	65.6	89.1	29.47s

From **Table 5**, we can see that the running time of the tree regularized classifier is not comparable with most of the classifiers, the reason is that the update of $\boldsymbol{\omega}$ in (13) is time-consuming, and the computational complexity of the proposed optimization algorithm is O(n^3). In the future, we will focus on the improvement of the efficiency of the optimization algorithm.

4.6 Comparison With The State Of The Art Methods

We finally compare the recognition accuracy of our method with the state of the art methods on the three datasets, the experimental results are shown in **Table 6**.

Table 6. Comparison of our method with the state of the art methods

Method	Year	Dataset		
		YouTube	Hollywood2	UCF50
Lu et al. [30]	2013	76.7	—	—
Wang et al. [9]	2013	85.4	59.9	85.6
Wang et al. [8]	2014	82.2	56.8	—
Beaudry et al. [34]	2016	—	—	88.3
Liu et al. [31]	2016	82.3	46.8	—
CNN [31]		79.8	45.3	—
DBN [31]		82.1	46.8	—
Liu et al. [35]	2016	94.4	—	86.5
Wang et al. [42]	2016	—	—	93.8
Kihl et al. [45]	2016	—	58.6	—
Our method		88.3	65.6	89.1

As shown in **Table 6**, we notice that our method has achieved better performance than most of the methods on the three datasets, owing that our method has captured the informative spatial and temporal configuration information of local features, and utilizes sparse coding to reduce quantization error in feature coding procedure. Moreover, the proposed tree regularized classifier boosts the performance further. Specifically, we list three popular deep learning methods [31, 35] shown in **Table 6**. Among them, the convolutional neural network (CNN) method [36] and the deep belief nets (DBN) method [37] are implemented by Liu et al. [31], we can see that the recognition accuracy achieved by the two methods is inferior to our method on YouTube and Hollywood2 datasets. Liu et al. [35] proposed the convolutional neural random fields and achieved the best recognition accuracy on the YouTube dataset.

As we can see from **Table 6**, the recognition accuracy we obtained may be not the highest. However, the main contribution of the paper is that we propose the baseline method to overcome the deficiencies of traditional BOVW, and we develop the tree regularized classifier to squeeze the performance further. Since our method is independent of the features, if we employ more advanced features, such as the features learned by “two-stream ”deep learning methods [44], it is possible to obtain higher recognition accuracy on the datasets.

5. Conclusion

In this paper, we combine sparse coding with spatio-temporal pyramid to recognize human action, and regard this method as the baseline. We find that there is a hierarchical structure in feature vector constructed by the baseline method. We encode the hierarchical structure into classifier through tree regularizer, and a tree regularized classifier is formulated by substituting L2 regularizer for tree regularizer in SVM. Besides, we present a method to solve the optimization problem of the proposed tree regularized classifier. We evaluate the proposed tree regularized classifier on YouTube, Hollywood2, and UCF50 datasets, the experimental results show that the recognition accuracy of the proposed tree regularized classifier is higher than SVM and other popular classifiers.

Generally, there are three research fields in human action recognition: feature extraction, feature representation and classification. Researchers usually focus on one of the fields in human action recognition, either feature extraction, or feature representation, or classification. To the best of our knowledge, few work address the structure information embeded in the

feature representation, and few work exploit the structure information to squeeze the classification performance. Inspired by the progress in natural language processing [13,16,17], we notice and take advantage of the structure information embeded in feature representation, and the method has been demonstrated in three action datasets. Therefore, the proposed method, especially the proposed tree regularized classifier gives a new insight into the enhancement of human action recognition accuracy, this is also the most important contribution of this paper.

References

[1] Wang X, Wang. L and Qiao. Y, "A comparative study of encoding, pooling and normalization methods for action recognition," in *Proc. of Asian Conference on Computer Vision*, 2012. Article (CrossRef Link).

[2] K. Yu, J. Yang, Y. Gong, "Linear Spatial Pyramid Matching Using Sparse Coding," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2009. Article (CrossRef Link).

[3] J.C. van Gemert, C.J. Veenman, A.W.M. Smeulders and J.M. Geusebroek, "Visual word ambiguity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp.1271-1283, 2010. Article (CrossRef Link).

[4] J. Wang, J. Yang, K. Yu, F. Lv and T. Huang, "Locality-constrained Linear Coding for Image Classification," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2010. Article (CrossRef Link).

[5] J. Sanchez, F. Perronnin, T. Mensink and J. Verbeek, "Image Classification with the Fisher Vector: Theory and Practice," *International Journal of Computer Vision*, vol. 105, no. 3, pp. 222-245, 2013. Article (CrossRef Link).

[6] A. Kovashka, K. Grauman, "Learning a hierarchy of discriminative space-time neighborhood features for human action recognition," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2010. Article (CrossRef Link).

[7] J. Wang, Z. Chen, Y. Wu, "Action recognition with multiscale spatio-temporal contexts," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2011. Article (CrossRef Link).

[8] H. Wang, C. Yuan, W. Hu, H. Ling, W. Yang, and C. Sun, "Action Recognition Using Nonnegative Action Component Representation and Sparse Basis Selection," *IEEE Transaction on Image Processing*, vol. 23, pp. 570-581, Feb. 2014. Article (CrossRef Link).

[9] H.Wang, A.Kläser, C.Schmid, C.Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International Journal of Computer Vision*, vol. 103, no. 1, pp. 60-79, 2013. Article (CrossRef Link).

[10] I. Laptev, M. Marszałek, C. Schmid, B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2008. Article (CrossRef Link).

[11] M. Ullah, SN. Parizi, I. Laptev, "Improving bag-of features action recognition with non-local cues," in *Proc. of British Machine Vision Conference*, 2010. Article (CrossRef Link).

[12] S. Lazebnik, C. Schmid, J. Ponce, "Beyond bags of features: Spatio-temporal pyramid matching for recognizing natural scene categories," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2006. Article (CrossRef Link).

[13] A. F. T. Martins, D. Yogatama, N.A. Smith and M. A. T. Figueiredo, "Structured Sparsity in Natural Language Processing: Models, Algorithms, and Applications," in *Proc. of the European Chapter of the Association for Computational Linguistics: Tutorials*, 2014. Article (CrossRef Link).

[14] M. Yuan, Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49-67, 2006. Article (CrossRef Link).

[15] P. Zhao, G. Rocha, B. Yu, "The composite absolute penalties family for grouped and hierarchical variable selection," *The Annals of Statistics*, vol. 37, no. 6A, pp. 3468-3497, 2009. Article (CrossRef Link).
[16] Yogatama. D, Smith. N. A, "Linguistic structured sparsity in text categorization," in *Proc. of the Annual Meeting of the Association for Computational Linguistics*, 2014. Article (CrossRef Link).
[17] Yogatama. D, Smith. N. A, "Making the most of bag of words: Sentence regularization with alternating direction method of multipliers," in *Proc. of the 31st International Conference on Machine Learning*, 2014. Article (CrossRef Link).
[18] L. Yan, W. Li, G. Xue, and D. Han, "Coupled Group Lasso for Web-Scale CTR Prediction in Display Advertising," in *Proc. of the 31st International Conference on Machine Learning*, 2014. Article (CrossRef Link).
[19] W. Deng, W. Yin, Y. Zhang, "Group sparse optimization by alternating direction method," in *Proc. of SPIE Optical Engineering+ Applications. International Society for Optics and Photonics*, 2013. Article (CrossRef Link).
[20] N. Parikh, S. Boyd, "Proximal algorithms," *Foundations and Trends in optimization*, vol. 1, no. 3, pp. 123-231, 2013. Article (CrossRef Link).
[21] Bach F, Jenatton R, Mairal J, Obozinski. G, "Optimization with sparsity-inducing penalties," *Foundations and Trends in Machine Learning*, vol, 1,no. 4, pp. 1-106, 2012. Article (CrossRef Link).
[22] Jenatton R, Mairal J, Obozinski G, Bach. F, "Proximal methods for hierarchical sparse coding," *The Journal of Machine Learning Research*, vol. 1,no. 12, pp. 2297-2334, 2011. Article (CrossRef Link).
[23] Qin Z, Goldfarb D, "Structured sparsity via alternating direction methods," *The Journal of Machine Learning Research*, vol.1, no. 13, pp. 1435-1468, 2012. Article (CrossRef Link).
[24] S. Boyd, N. Parikh, E. Chu, B. Peleato, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol.3 ,no. 1, pp. 1-122, 2011. Article (CrossRef Link).
[25] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2009. Article (CrossRef Link).
[26] J. Liu, J. Luo and M. Shah, "Recognizing realistic actions from videos "in the wild"," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2009. Article (CrossRef Link).
[27] J. Platt, "Fast training of support vector machines using sequential minimal optimization," *Advances in kernel methods—support vector learning*, vol. 3, no. 1, pp. 32-37, 1999. Article (CrossRef Link).
[28] C. Chang and C. Lin, "LIBSVM : a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1-27, 2011. Article (CrossRef Link).
[29] Duda. R, Hart. P, Stork. D, *Pattern classification*, 2nd. Ed, John Wiley & Sons, New York, 2012. Article (CrossRef Link).
[30] Z. Lu and Y. Peng, "Latent semantic learning with structured sparse representation for human action recognition," *Pattern Recognition*, vol. 46, no. 7, pp. 1799-1809, 2013. Article (CrossRef Link).
[31] L. Liu, L. Shao, X. Li and K. Lu, "Learning Spatio-Temporal Representations for Action Recognition: A Genetic Programming Approach," *IEEE Transactions on Cybernetics*, vol. 46, no. 1, pp. 158-170. 2016. Article (CrossRef Link).
[32] Kishore K. Reddy, and Mubarak Shah, "Recognizing 50 Human Action Categories of Web Videos," *Machine Vision and Applications*, vol. 24, no. 5, pp. 971-987. 2013. Article (CrossRef Link).
[33] Y G. Jiang, Dai Q, Liu W, X Y Xue, and C W .NGO, "Human Action Recognition in Unconstrained Videos by Explicit Motion Modeling," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3781-3795. 2015. Article (CrossRef Link).
[34] C. Beaudry, R. Péteri, and L Mascarilla, "An efficient and sparse approach for large scale human action recognition in videos," *Machine Vision and Applications*, vol. 27, no. 4, pp. 529-543. 2016. Article (CrossRef Link).

[35] C. Liu, J. Liu, Z. He, Y. Zhai, Q. Hu, and Y Huang, "Convolutional neural random fields for action recognition," *Pattern Recognition*, vol. 59, pp. 213-224. 2016. Article (CrossRef Link).
[36] M. Ranzato, F. Huang, Y. Boureau, and Y. LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2007. Article (CrossRef Link).
[37] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006. Article (CrossRef Link).
[38] J. Mairal, F. Bach and J. Ponce, "Sparse Modeling for Image and Vision Processing," *Foundations and Trends in Computer Graphics and Vision*, vol. 8, no.2-3, pp. 85-283, 2014. Article (CrossRef Link).
[39] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no.2, pp:210–227, 2009. Article (CrossRef Link).
[40] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?," in *Proc. of International Conference on Computer Vision*, pp: 471-478, 2011. Article (CrossRef Link).
[41] R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of machine learning research*, vol.9 no.8, pp:1871-1874, 2008. Article (CrossRef Link).
[42] L. Wang, Y. Qiao, and X. Tang, "MoFAP: A Multi-Level Representation for Action Recognition," *International Journal of Computer Vision*, vol 119, no.3, pp.254-271, 2016. Article (CrossRef Link).
[43] S. J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "A interior-point method for large-scale l1-regularized least squares," *IEEE Journal on Selected Topics in Signal Processing*, vol 1, no.4, pp: 606–617, 2007. Article (CrossRef Link).
[44] K. Simonyan, A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *Advances in Neural Information Processing Systems*, 2014. Article (CrossRef Link).
[45] O. Kihl, D. Picard, and P.H. Gosselin, "Local polynomial space–time descriptors for action classification," *Machine Vision and Applications*, vol.27, no.3, pp: 351-361, 2016. Article (CrossRef Link).

Huiwu Luo was born in 1986. He is currently a PH.D candidate at National Key Laboratory of Automatic Target Recognition (ATR), School of Electronic Science and Engineering, National University of Defense Technology. His research interests include image classification, action recognition, and machine learning.

Fei Zhao was born in 1983. He is currently an assistant professor at National Key Laboratory of Automatic Target Recognition (ATR), School of Electronic Science and Engineering, National University of Defense Technology. His research interests include object detection, image processing, and automatic target recognition.

Shangfeng Chen was born in 1978. He is currently an assistant professor at National Key Laboratory of Automatic Target Recognition (ATR), School of Electronic Science and Engineering, National University of Defense Technology. His research interests include automatic target recognition and target tracking.

Huanzhang Lu was born in 1963. He is currently a professor at National Key Laboratory of Automatic Target Recognition (ATR), School of Electronic Science and Engineering, National University of Defense Technology. His research interests include automatic target recognition, real time system, and computer vision system.