Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

sNN-LDS: Spatio-temporal Non-negative Sparse Coding for Human Action Recognition

Thomas Guthier, Adrian Sosic, Volker Willert, Julian Eggert

2013

Preprint:

This is an accepted article published in Artificial Neural Networks andProceedings of the 24th International Conference on Artificial Neural Networks (ICANN). The final authenticated version is available online at: https://doi.org/[DOI not available]

sNN-LDS: Spatio-temporal Non-negative Sparse Coding for Human Action Recognition

Thomas Guthier[1], Adrian Šošić[2], Volker Willert[1], and Julian Eggert[3]

[1]Control Theory and Robotics, TU Darmstadt, Landgraf-Georg Strasse 4, Darmstadt, Germany
[2]Signal Processing Group, TU Darmstadt, Merckstrasse 25, Darmstadt, Germany
[3]Honda Research Institute Europe, Carl-Legien Strasse 30, Offenbach, Germany

Abstract. Current state-of-the-art approaches for visual human action recognition focus on complex local spatio-temporal descriptors, while the spatio-temporal relations between the descriptors are discarded. These bag-of-features (BOF) based methods [8, 9, 11, 12] come with the disadvantage of limited descriptive power, because class-specific mid- and large-scale spatio-temporal information, such as body pose sequences, cannot be represented. To overcome this restriction, we propose sparse non-negative linear dynamical systems (sNN-LDS) as a dynamic, parts-based, spatio-temporal representation of local descriptors. We provide novel learning rules based on sparse non-negative matrix factorization (sNMF) [7] to simultaneously learn both the parts as well as their transitions. On the challenging UCF-Sports dataset [2] our sNN-LDS combined with simple local features is competitive with state-of-the-art BOF-SVM methods [12].

1 Introduction

Visual human action recognition is a vivid research topic where the goal is to classify various actions performed by humans in a video. Actions consist of temporal sequences of body poses; and single poses, such as standing, can be part of more than one action class, *e.g.* golfing or kicking. To robustly separate different action classes it is thus useful to encode the temporal relations in addition to static pose information.

Human actions are restricted by the human body structure, which reduces the space of possible poses significantly. However, human actions can vary in speed, intensity and individual performance, alongside with additional variations, such as different view points, scales and occlusions. Current approaches combine the ideas of local spatio-temporal histogram descriptors with *bag-of-features* (BOF) and *support vector machine* (SVM) classifiers [8, 9, 11, 12]. Here, the spatio-temporal information is encoded in the local descriptors, which are typically restricted to 3D cubes around salient key points [11] or dense trajectories [12]. Each action video is then represented by its descriptors mapped onto a *codebook* that can be pre-learned *e.g.* with k-means clustering, sparse coding [3, 9] or non-negative matrix factorization [4, 8]. The main downside of the BOF-methods is that the spatio-temporal relations between the local descriptors are discarded. Consequently, all topological information as well as temporal relations between the descriptors are lost for the classification. Due to the local nature of the descriptors it is thus not possible to describe action specific body poses as an entity and subsequently, it is not possible to explicitly describe pose-sequences either.

In related work, *e.g.* [15] the inter-descriptor spatio-temporal relations are described by designed and locally bound contextual features. A similar approach, *i.e.* using designed relations, is proposed in [14].

Unlike the related approaches we do not hand design, but *learn* the spatio-temporal relations between the local descriptors and represent them as *sparse non-negative linear dynamical systems* (sNN-LDS). The local descriptors are grouped into a pooling block structure, similar to the complex cells of feed-forward-neural-networks. The spatial relations between the descriptors are thus encoded by their relative positions in the pooling block grid. The sNN-LDS models the input consisting of all descriptors of the grid by a linear superposition of local prototypical parts while the temporal relations between consecutive video frames are captured by the dynamics of the sNN-LDS.

In contrast to other learning algorithms, *e.g.* fused lasso [16] or [13], in our method both, the transition matrix $\mathcal{K}$, that defines the model dynamics, and the observation matrix $\mathcal{W}$, which consists of prototypical parts, are learned. We use fast and simple update rules based on *sparse non-negative matrix factorization* (sNMF) [7]. The algorithm can either be interpreted as incorporating non-negativity and sparsity constraints into high-dimensional linear dynamical systems or as an extension of sNMF with a novel transition component that models the temporal relations between the activations. The sparse and parts-based properties give rise to models that are more robust than holistic or designed models. Compared to classic BOF approaches, which make use of high-dimensional spatio-temporal relations *inside* a space-time-volume, we propose large scale spatial and temporal relations *between* simple low-dimensional local descriptors.

Next, we introduce the central part of our approach, the sNN-LDS algorithm. Thereafter, we describe our classification system for human action recognition based on the sNN-LDS and provide benchmark results on the Weizmann [1] and UCF-Sports datasets [2].

2 Sparse Non-negative Linear Dynamical System (sNN-LDS)

Desired properties of a learned model are: a) the ability to *explain the given data* and b) that it should *generalize to new data* while staying *class-specific*. This can be realized by *learning and representing* all relevant variations in one global model. In order to obtain such a representation we choose a dynamic, generative, linear and parts-based approach, because local parts tend to generalize better than holistic models, while the assumption of an underlying generative process guarantees the entire input to be represented. We achieve parts-basedness by combining the ideas of *sparse coding* [3] and *non-negative matrix factorization* (NMF) [4]. Non-negativity ensures that components can only be added, while sparsity prefers as few components as necessary, meaning that from the set of possible solutions those with few but meaningful parts are favoured. The temporal relations are modeled by the transitions between the activations of the parts.

2.1 sNN-LDS Model

A sNN-LDS is a generative model with non-negativity constraints on all model parameters that uses sparse activations for the encoding of the input. The model is depicted

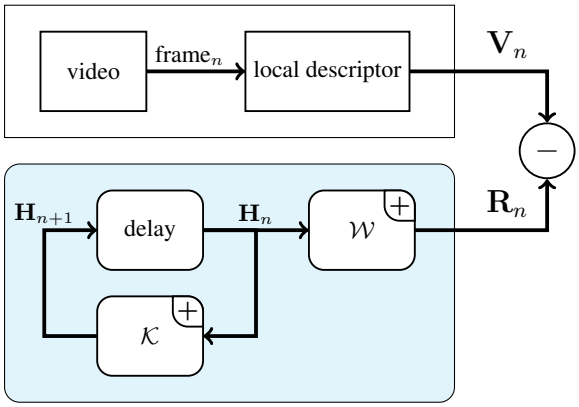

Fig. 1. The sNN-LDS (blue) works as a generative model whose internal state $\mathbf{H}_n$ is adapted so that the model output $\mathbf{R}_n$ resembles the observed descriptors $\mathbf{V}_n$.

in Fig. 1. Given a set $\mathcal{V} \in \mathbb{R}^{X \times N}$ of input signals $\mathbf{V}_n \in \mathbb{R}^X$ ($X :=$ input dimension, $N :=$ number of inputs) the corresponding observation model or reconstruction $\mathbf{R}_n$ is given by a weighted sum of normalized basis vectors $\bar{\mathcal{W}} \in \mathbb{R}^{X \times J}$ ($J :=$ number of basis vectors), with $\bar{\mathbf{W}}_j = \frac{\mathbf{W}_j}{\|\mathbf{W}_j\|_2}$,

$$\mathbf{V}_n \approx \mathbf{R}_n = \sum_j h_{jn} \bar{\mathbf{W}}_j, \tag{1}$$

$$v_{xn}, h_{jn}, w_{xj} \geq 0, \quad \forall n \in [1, N], x \in [1, X], j \in [1, J]. \tag{2}$$

The weight or activation matrix $\mathcal{H} \in \mathbb{R}^{J \times N}$ consists of activation vectors $\mathbf{H}_n$. The system dynamics are represented by a non-negative linear transition model $\mathcal{K} \in \mathbb{R}^{J \times J}$ that encodes the relations between the activation vectors $\mathbf{H}_n$ and $\mathbf{H}_{n+1}$ of all consecutive inputs $\mathbf{V}_n$ and $\mathbf{V}_{n+1}$. The predicted activations are

$$\hat{h}_{jn} = \sum_l k_{jl} h_{ln-1}, \quad \forall n \in [2, N], j \in [1, J], \tag{3}$$

$$\hat{\mathcal{H}} = \mathcal{K}\mathcal{H}\mathcal{S}, \tag{4}$$

$$k_{jl} \geq 0, \quad \forall j \in [1, J], l \in [1, J], \tag{5}$$

with the shift matrix $\mathcal{S} = \begin{pmatrix} 0 & \mathcal{I} \\ 0 & 0 \end{pmatrix}, \mathcal{S} \in \mathbb{R}^{N \times N}$.

2.2 Learning the sNN-LDS Model

Our learning method is motivated by the sNMF learning algorithm. Here, the unknown model parameters $\mathcal{W}$ and $\mathcal{H}$ are learned by minimizing the reconstruction energy

$$E_r = \frac{1}{2}\|\mathcal{V} - \bar{\mathcal{W}}\mathcal{H}\|_F^2 + \lambda_H \|\mathcal{H}\|_1 \tag{6}$$

with respect to randomly initialized matrices $\mathcal{H}$ and $\mathcal{W}$ via iterative, multiplicative gradient descent. The sNMF model, however, assumes that the inputs $\mathbf{V}_n$ are conditionally independent. In order to account for these dependencies we introduce the following transition energy into our sNN-LDS model.

$$E_t = \frac{1}{2}\sum_{n=2} \|\mathbf{V}_n - \bar{\mathcal{W}}\mathcal{K}\mathbf{H}_{n-1}\|_2^2, \tag{7}$$

$$= \frac{1}{2}\|\mathcal{V}\mathcal{Q} - \bar{\mathcal{W}}\mathcal{K}\mathcal{H}\mathcal{S}\|_F^2, \tag{8}$$

with $\mathcal{Q} = \begin{pmatrix} 0 & 0 \\ 0 & \mathcal{I} \end{pmatrix}$, $\mathcal{Q} \in \mathbb{R}^{N\times N}$, that masks out the first input $\mathbf{V}_1$. Since the activations may be high-dimensional compared to the amount of training data, we add a regularization parameter, *i.e.* enforce sparsity on the transitions. The overall energy function of the sNN-LDS then becomes

$$\begin{aligned} E =& \frac{1}{2}\|\mathcal{V} - \bar{\mathcal{W}}\mathcal{H}\|_F^2 + \frac{1}{2}\|\mathcal{V}\mathcal{Q} - \bar{\mathcal{W}}\mathcal{K}\mathcal{H}\mathcal{S}\|_F^2 \\ &+ \lambda_H\|\mathcal{H}\|_1 + \lambda_{KS}\|\mathcal{K}\|_1. \end{aligned} \tag{9}$$

Following the concept of sNMF, the unknown model parameters $\mathcal{W}$ and $\mathcal{K}$ as well as the activations $\mathcal{H}$ of the sNN-LDS are learned by minimizing the energy function (9) with respect to randomly initialized matrices $\mathcal{W}$, $\mathcal{K}$ and $\mathcal{H}$ via iterative, multiplicative gradient descent. The update rules are

$$\mathcal{H} \to \mathcal{H} \circ \frac{(\nabla_{\mathcal{H}} E)^-}{(\nabla_{\mathcal{H}} E)^+}, \tag{10}$$

$$\mathcal{W} \to \mathcal{W} \circ \frac{(\nabla_{\mathcal{W}} E)^-}{(\nabla_{\mathcal{W}} E)^+}, \tag{11}$$

$$\mathcal{K} \to \mathcal{K} \circ \frac{(\nabla_{\mathcal{K}} E)^-}{(\nabla_{\mathcal{K}} E)^+}, \tag{12}$$

where the positive and negative gradient components, including the inner derivative of the normalized basis vectors [7], are given as

$$(\nabla_{\mathcal{H}} E)^+ = \bar{\mathcal{W}}^\top\bar{\mathcal{W}}\mathcal{H} + \mathcal{K}^\top\bar{\mathcal{W}}^\top\bar{\mathcal{W}}\mathcal{K}\mathcal{H}\mathcal{S}\mathcal{S}^\top + \lambda_H, \tag{13}$$

$$(\nabla_{\mathcal{H}} E)^- = \bar{\mathcal{W}}^\top\mathcal{V} + \mathcal{K}^\top\bar{\mathcal{W}}^\top\mathcal{V}\mathcal{S}^\top, \tag{14}$$

$$(\nabla_{\bar{\mathcal{W}}} E)^+ = \bar{\mathcal{W}}\mathcal{H}\mathcal{H}^\top + \bar{\mathcal{W}}\mathcal{K}\mathcal{H}\mathcal{S}\mathcal{S}^\top\mathcal{H}^\top\mathcal{K}^\top, \tag{15}$$

$$(\nabla_{\bar{\mathcal{W}}} E)^- = \mathcal{V}\mathcal{H}^\top + \mathcal{V}\mathcal{Q}\mathcal{S}^\top\mathcal{H}^\top\mathcal{K}^\top, \tag{16}$$

$$(\nabla_{\mathcal{W}} E)^+ = (\nabla_{\bar{\mathcal{W}}} E)^+ + \bar{\mathcal{W}}\bar{\mathcal{W}}^\top(\nabla_{\bar{\mathcal{W}}} E)^-, \tag{17}$$

$$(\nabla_{\mathcal{W}} E)^- = (\nabla_{\bar{\mathcal{W}}} E)^- + \bar{\mathcal{W}}\bar{\mathcal{W}}^\top(\nabla_{\bar{\mathcal{W}}} E)^+, \tag{18}$$

$$(\nabla_{\mathcal{K}} E)^+ = \bar{\mathcal{W}}^\top\bar{\mathcal{W}}\mathcal{K}\mathcal{H}\mathcal{S}\mathcal{S}^\top\mathcal{H}^\top + \lambda_{KS}, \tag{19}$$

$$(\nabla_{\mathcal{K}} E)^- = \bar{\mathcal{W}}^\top\mathcal{V}\mathcal{Q}\mathcal{S}^\top\mathcal{H}^\top. \tag{20}$$

Throughout all experiments the input is normalized using the max-norm and the regularization parameters are set to $\lambda_H = 0.1$ and $\lambda_{KS} = 0.2$.

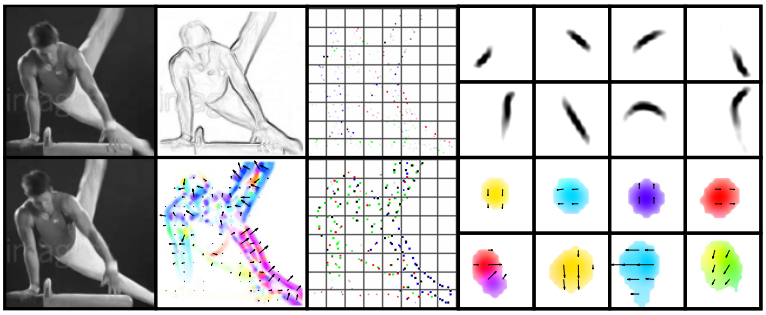

Fig. 2. Gradient and optical flow descriptors. Upper row from left to right: input image $\mathbf{I}_n$, gradient amplitude $\mathbf{V}_n^{(g)}$, simple cell response $\mathbf{S}_n^{(g)}$ with pooling cell structure and the eight learned gradient patterns $\mathcal{W}^{(g)}$ (scaled by factor 4). Lower row from left to right: input image $\mathbf{I}_{n+1}$, optical flow $\mathbf{V}_n^{(f)}$, simple cell response $\mathbf{S}_n^{(f)}$ with pooling cell structure and the eight learned optical flow patterns $\mathcal{W}^{(f)}$ (scaled by factor 4).

3 System Outline

In the following we explain how the sNN-LDS is included in our human action recognition algorithm. Similar to a BOF approach our algorithm consists of four components: 1.) Figure centric dense sampling, 2.) Calculation of local descriptors, 3.) Mapping local descriptors onto a sNN-LDS model and 4.) Frame-wise classification of the activations.

3.1 Figure Centric Dense Sampling

For each frame we cut out a window around the person whose action we want to classify and rescale it to a reference size of 128×128 pixels, resulting in a *figure centric* representation.[1] In this window we perform a dense sampling with 50% overlapping pooling blocks, each having a spatial resolution of 32×32 pixels which leads to $7 \cdot 7 = 49$ blocks. For all blocks we calculate low level features of gradient amplitudes and optical flow fields estimated using the algorithm described in [10].

3.2 Local Descriptors

For each pair of input images $\mathbf{I}_n$ and $\mathbf{I}_{n+1}$ the local descriptors are biologically inspired simple cell/complex cell descriptors of the gradient amplitude $\mathbf{V}_n^{(g)}$ and the optical flow field $\mathbf{V}_n^{(f)}$. To this end, a set of eight basic patterns is learned using the unsupervised learning algorithm described in [6]. The patterns and the simple cell responses are shown in Fig. 2. The simple cell response $\mathbf{s}_{dn}^{(g)}, d \in [1, ..., 8]$ of the gradient patterns $\mathbf{W}_d^{(g)}$ for the input $\mathbf{V}_n^{(g)}$ is given by

$$\mathbf{s}_{dn}^{(g)} = \text{corr}_2(\mathbf{V}_n^{(g)}, \mathbf{W}_d^{(g)}), \tag{21}$$

[1] This step is currently performed manually but could be automized by a robust motion detector, *e.g.* [5], in future work.

with the two dimensional correlation corr_2. The complex cell response is an overlapping pooling operation

$$\hat{c}_{dn}^{(g)}(\mathbf{x}) = \sum_{\mathbf{y} \in A(\mathbf{x})} \mathbf{s}_{dn}^{(g)}(\mathbf{y}), \tag{22}$$

and the pooled values are subsequently binarized

$$c_{dn}^{(g)}(\mathbf{x}) = 1, \quad \text{if} \quad \hat{c}_{dn}^{(g)}(\mathbf{x}) > 0.2, \tag{23}$$

$$c_{dn}^{(g)}(\mathbf{x}) = 0, \quad \text{else.} \tag{24}$$

The same procedure is performed for the optical flow patterns. For each of the 49 blocks we get a local descriptor that contains the complex cell responses for all 16 basic patterns. The final dimension of the descriptor grid $\mathbf{V}_n$ is $D = 49 \cdot 16 = 784$ for each image. An input image $\mathbf{I}_n$ is therefore described by its local descriptor grid $\mathbf{V}_n \in \mathbb{R}^D$ and the spatial relations between the local descriptors are captured by the relative positions of the pooling blocks.

3.3 Learning the sNN-LDS

We combine the local descriptors of a batch of videos to create the input matrix $\mathcal{V}$, for which the parameters $\mathcal{W}$, $\mathcal{K}$ and $\mathcal{H}$ of a sNN-LDS are learned using the update rules in eq. (10), (11) and (12). After the model parameters are learned, $\mathcal{W}$ and $\mathcal{K}$ are fixed. For each new video, the activations $\mathcal{H}$ are calculated using eq. (10).

3.4 Frame-wise Classification

Each frame $\mathbf{V}_n$ of an input video $\mathcal{V}$ is represented by the corresponding activation vector $\mathbf{H}_n$, which defines the current state of the sNN-LDS. Our sNN-LDS can thus be considered as a *codebook* for which the basis vectors $\mathcal{W}$ and the corresponding transitions $\mathcal{K}$ define the prototypical words, while the activations $\mathbf{H}_n$ represent the presence of the words in the current frame.

The feature vectors of the training videos and the mirrored version of the training videos are then used to train a soft-margin multiclass SVM with radial basis function kernels and class-weights that account for unbalanced training sets. All classification parameters are learned by 5-fold cross-validation on the training data. The classification of the test data is performed per frame and the video result is thereafter the weighted classification result of all its frames. The weighting factor for each frame and class is provided by the SVM classifier.

4 Classification Results

We evaluate how the sNN-LDS perform in leave-one-out experiments on the challenging 9-class UCF-Sports dataset and the 10-class Weizmann dataset. On the UCF-Sports dataset sequences of the same action, *e.g.* kicking, include karate kicks as well as soccer

kicks which differ strongly in their movements. The evaluation focuses on two aspects: First, the influence of the transition matrix $\mathcal{K}$, *i.e.* the difference between sNMF and sNN-LDS. Second, we compare the classification results for two different numbers of basis vectors J.

	sNMF		sNN-LDS		Related Work		
J	50	100	50	100	[5]	[12]	[17]
Weizmann	0.98	0.99	0.99	**1.00**	**1.00**	-	-
UCF-Sports	0.88	0.87	0.90	**0.92**	-	0.89	0.90

Table 1. Classification results for leave-one-out experiments.

Table 1 shows the results for the different experiments on the UCF-Sports and Weizmann dataset. The sNN-LDS slighty outperforms the sNMF on the Weizmann datasets and by 3% on the UCF-Sports datasets, which shows that modeling the temporal relations improves the classification performance. Increasing the number of basis vectors J from 50 to 100 improves the results, but not as significantly as adding the temporal relations. Our algorithm outperforms the state-of-the-art algorithm proposed in [12]. However, as discussed in [17], we would expect their results to be competitive with ours when applied to the same figure-centric representation.

5 Summary & Discussion

We show that our generative sNN-LDS, which explicitly represents the spatio-temporal relations between the local descriptors, is a highly discriminative dynamical model that is well suited for, yet not restricted to, human action recognition. Due to its non-negative representation, learning rules from sNMF can be adapted to simultaneously learn all model parameters of the sNN-LDS.

Our experiments on the UCF-Sports dataset raise the following question: Are the large scale spatio-temporal relations *between* the local descriptors more important than complex spatio-temporal relation *inside* a local descriptor?

The sNN-LDS system slightly outperforms the BOF methods [12, 17] by 2% even though the dimensionality of their highly sophisticated trajectory-descriptors [12] is as high as 396, while our local descriptors have only 16 dimensions. In addition, the dimensionality of our sNN-LDS is significantly smaller than the codebooks learned with k-means in *e.g.* [12, 17], which typically consist of 4000 clusters, while we use only 100 basis vectors. Hence, we assume that the descriptive power of the sNN-LDS must come from the spatial and temporal relations between the simple local descriptors.

References

1. M. Blank, L. Gorelick, E. Shechtman, M. Irani and R. Basri, Actions as Space-Time Shapes, *IEEE Int. Conf. on Computer Vision* (ICCV), 2005.
2. M.D. Rodriguez, J.Ahmed, and M.Shah, "Action MACH: A Spatio-temporal Maximum Average Correlation Height Filter for Action Recognition.", *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.
3. B. Olshausen and D.J. Field, Emergence of simple-cell receptive field properties by learning a sparse code for natural images, *Nature*, vol. 381, pp. 607-609, 1996.
4. D.D. Lee and S. Seung, Learning the parts of objects by non-negative matrix factorization, *Nature*, vol. 401, pp. 788-791, 1999.
5. Y.Tian and R.Sukthankar and M.Shah, "Spatiotemporal Deformable Part Models for Action Detection", *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013.
6. T. Guthier, J. Eggert and V. Willert, "Unsupervised learning of motion patterns", *European Symposium on Artificial Neural Networks* (ESANN), 2012.
7. J. Eggert and E. Koerner, "Sparse coding and NMF", *IEEE Int. Joint Conf. on Neural Networks* (IJCNN), vol. 4, pp. 2529-2533, 2004.
8. S.M.Amiri, P.Nasiopoulos, and V.Leung. "Non-negative sparse coding for human action recognition." *IEEE Int. Conf. on Image Processing (ICIP)*, 2012.
9. T.Guha, and R.K.Ward. "Learning sparse representations for human action recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34.8 (2012): 1576-1588.
10. T.Guthier, V.Willert, A.Schnall, K.Kreuter, and J.Eggert, "Non-negative sparse coding for motion extraction", *IEEE Int. Joint Conf. on Neural Networks* (IJCNN), 2013.
11. Wang, H., Ullah, M. M., Klaser, A., Laptev, I., and Schmid, C., "Evaluation of local spatio-temporal features for action recognition." *British Machine Vision Conference (BMVC)*, 2009.
12. Wang, H., Klaser, A., Schmid, C., and Liu, C. L., "Dense trajectories and motion boundary descriptors for action recognition." *International Journal of Computer Vision*: 1-20, 2013.
13. B. Lakshminarayanan, and R. Raich, "Non-negative matrix factorization for parameter estimation in hidden markov models." *IEEE Int. Workshop on Machine Learning for Signal Processing (MLSP)*, 2010.
14. P. Bilinski and F. Bremond, "Contextual statistics of space-time ordered features for human action recognition", *IEEE Int. Conf. on Advanced Video and Signal-Based Surveillance* (AVSS), pp. 228–233, 2012.
15. J. Wang, Z. Chen and Y. Wu, "Action recognition with multiscale spatio-temporal contexts", *IEEE Conf. on Computer Vision and Pattern Recognition* (CVPR), pp. 3185–3192, 2011.
16. Tibshirani, R., Saunders, M., Rosset, S., Zhu, J. and Kneight, K., "Sparsity and smoothness via the fused lasso" *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*: 91–108, 2005.
17. Klaser, A., Marszałek, M., Laptev, I., Schmid, C. and others, "Will person detection help bag-of-features action recognition? ", 2010.