Image and Vision Computing 39 (2015) 38–51

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Non-negative matrix completion for action detection ☆

Ehsan Adeli-Mosabbeb [a,1,2,*], Mahmood Fathy [a,2,3]

[a] Computer Engineering Department, Iran University of Science and Technology, Narmak, Tehran 16486–13114, Iran

ARTICLE INFO

Article history:
Received 28 December 2012
Received in revised form 16 February 2015
Accepted 23 April 2015
Available online 15 May 2015

Keywords:
Matrix completion
Multi-label classification
Weakly supervised classification
Human activity recognition
Alternating direction method
Convex optimization

ABSTRACT

With the increasing number of videos all over the Internet and the increasing number of cameras looking at people around the world, one of the most interesting applications would be human activity recognition in videos. Many researches have been conducted in the literature for this purpose. But, still recognizing activities in a video with unrestricted conditions is a challenging problem. Moreover, finding the spatio-temporal location of the activity in the video is another issue. In this paper, we present a method based on a non-negative matrix completion framework, that learns to label videos with activity classes, and localizes the activity of interest spatio-temporally throughout the video. This approach has a multi-label weakly supervised setting for activity detection, with a convex optimization procedure. The experimental results show that the proposed approach is competitive with the state-of-the-art methods.

1. Introduction

Action detection and recognition has many applications, including vision based surveillance, human–computer interaction, patient monitoring systems and a lot more [1,2]. This makes it a very important field in the computer vision studies, today. Understanding the behavior of an individual in a video sequence is a challenging task due to several different issues, including the large variability in the imaging conditions as well as the way different people perform a particular action, while in the meantime the background clutter and motion make the problem of extracting information from a human action rather difficult. Furthermore, the high dimensionality of such data is another significant challenge for these recognition problems.

The problem of action detection is comprised of two subproblems, recognition and localization. Traditional approaches use fully annotated video datasets for the process of learning, where each video is labeled with an activity class and the activity location is defined, usually via bounding boxes for in each individual video frame. But it is very hard to provide ground truth data which labels every individual action in the video sequence, with bounding boxes for the action in every frame. Thus, an approach which can recognize actions in videos and extract its spatio-temporal location is of great interest. For this purpose, we consider a weakly supervised setting, where instead of labeling the manually annotated spatio-temporal locations (bounding boxes), we label each video with one or more particular action classes. With this formulation, we will be dealing with only positive and negative videos, for each action. Negative videos are those which do not have any instances of the activity of interest. On the other hand, positive videos of a particular activity contain the activity of interest somewhere in their sequence of frames, but we do not know where. Since the supervision is weak, providing datasets for training would be a simple task, whereas, the learning task would be a challenging one.

One might notice that this problem is very similar to the formulation of a Multiple-Instance Learning (MIL) problem [3–6]. In MIL, the learning task is to learn a concept to recognize positive instances in a bunch of positive and negative bags. Negative bags contain all negative instances, while positive bags contain at least one positive instance. The objective would be to train from previously labeled positive and negative bags to find positive instances in both train and test bags. For our purpose, we are also dealing with videos which may or may not contain an activity of interest. Accordingly, we could model the videos as positive or negative bags. Nonetheless, this could not be done so easily, and the problem is not equivalent to one of a MIL method. This is because a video sequence is a single entity and is not composed of a bunch of instances, in which the activity or activities of interest happen.

In order to model our problem in a MIL framework, we use a simple technique to break a video down to several potential activity regions and the rest as the background. We treat a video as a vector of quantized features, similar to the bag of words (BoWs) model [7]. During the recognition procedure we correct the representative feature vector of each video such that the non-activity regions are taken out. This is done via rank minimization criteria over the features matrix. This framework

☆ This paper has been recommended for acceptance by Ahmed Elgammal.

* Corresponding author.
E-mail addresses: eadeli@iust.ac.ir (E. Adeli-Mosabbeb), mahfathy@iust.ac.ir (M. Fathy).
[1] Tel.: +1 4122568280; fax: +98 2173225322.
[2] Tel.: +98 2173225308; fax: +98 2173225322.
[3] Tel./fax: +1 4162228676.

http://dx.doi.org/10.1016/j.imavis.2015.04.006
0262-8856/

Fig. 1. Samples of video frames from Hollywood Human Action (HOHA) dataset, with potential activity regions (yellow rectangles) and the selected region as the activity of interest (green rectangle).

learns a latent representation for each activity in the whole dataset and finds the best action label(s) for each test video. A simple search throughout the potential spatio-temporal regions in the video, can find us the location of the activity of interest in space and time. Fig. 1 shows some sample frames from the Hollywood Human Action (HOHA) dataset, where potential activity regions are marked using yellow bounding boxes. The selected region for the activity of interest in depicted in green. Fig. 2 illustrates the process of activity detection, presented in this paper. We provide our learner with a number of videos, each of which has been weakly labeled with one or a number of actions. The testing procedure would be defined as determining the label(s) for each unlabeled video, along with finding the spatio-temporal region(s) of the activity or activities. We test our approach on five well-known activity recognition datasets: KTH, Weizmann, MSR2, HOHA and UCF Sports.

As also discussed earlier, activity recognition/localization is a hard task due to the clutter and the noise from the background and/or the imaging conditions, besides the variability in performing the actions by the subjects. Many previous works have targeted activity recognition [8–12] and localization [13–17], but few works have proposed methods to solve both, simultaneously [18–24]. Most approaches use fully annotated datasets and train a recognition/localization framework in fully supervised settings. In order to address the above issues different features have been introduced [8,25], interest region detectors such as space–time volumes [26] or trajectories [27,10] have been used and different classifiers are utilized [28,25]. These methods have improved recognition results, but they often do not incorporate spatial and temporal relationships between regions of interest in videos. Hence, recognizing and localizing activities as a joint process in weakly supervised settings could improve both recognition and localization results.

We observe that joint recognition and localization of human activities in a weakly supervised setting is a very good application for MIL, since labeling videos and annotating the activity in every single frame is a very arduous task. On the other hand, each video in the dataset could be treated as a positive or negative bag (containing or not contracting an activity of interest). We develop a MIL model, based on low-rank matrix completion, where the features vector for each video is polished to take the background context and non-activity related regions effects out. Thence, we will be able recover a representative feature vector for each single activity in the dataset in a convex multi-label setting. Furthermore, we use a number of fixed length histogram of densely sampled features, which captures both the visual content of the scene and the temporal changes in the scene. Therefore, our method is mostly independent from the video content, view point and the background/imaging conditions, to some extent.

Our contributions could be summarized as: (1) developing a multi-label recognition framework with a convex optimization process for the problem of activity recognition, in which we use the well known matrix completion to recover the labels for the test videos. Each video may have one or more activities of interest occurring in different spatio-temporal segments. (2) Using the histograms of densely sampled features throughout the video and correcting the histograms such that for each class a representative histogram is extracted. This histogram could be searched in the video over the potential locations of the activity, to localize the activity, spatio-temporally. (3) Proposing a new formulation for matrix completion to deal with recognition/localization in video.

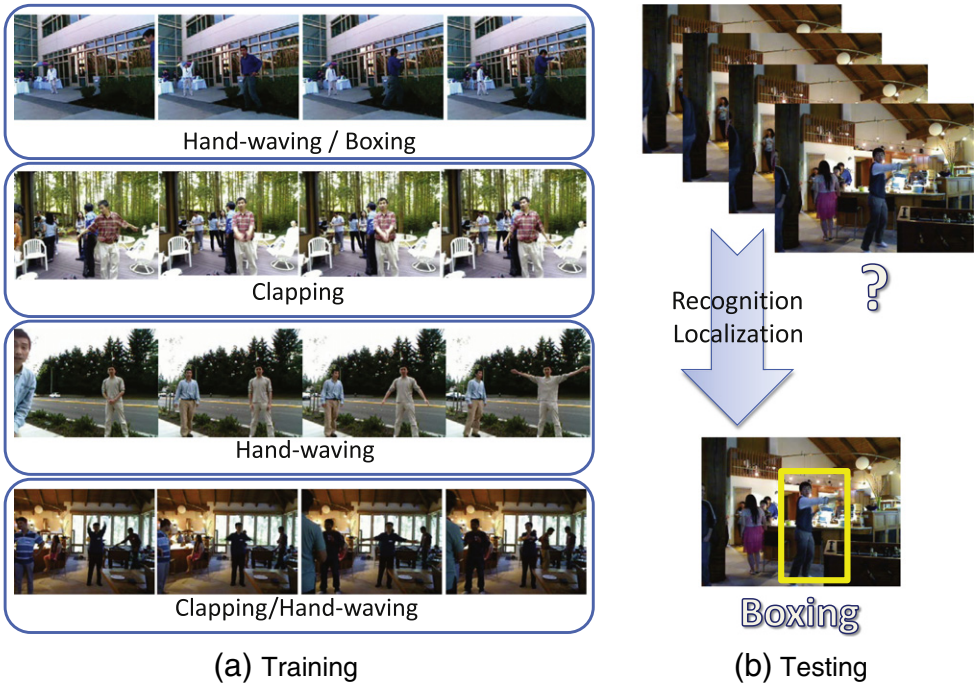


Fig. 2. For training, we provide our weakly supervised learner with a bunch videos which are labeled with one or more activities. In order to test unseen videos, our learner is provided with the video features. It labels the test videos and finds the spatio-temporal location of the activity/activities of interest, throughout the video.

(4) Developing an activity recognition system in a weakly supervised multi-label setting. (5) Developing a non-negative matrix completion framework based on Alternating Direction Method (ADM) of multipliers.

The rest of the paper is organized as follows: Section 2 reviews the literature of the work and Section 3 explains the notations used in this paper. Section 4 discusses conventional matrix completion techniques. Section 5 explains the proposed approach in details and Section 6 gives the results on some real world datasets. Finally, Section 7 concludes the paper.

2. Related works

Multiple-instance learning dates back to [29] for solving drug activity prediction problem with axis-parallel rectangles. Following that a simple general framework was proposed by [3], in which a probabilistic measure for diverse density is proposed. A search over the diverse density surface and finding the global maxima would lead to the solution to the multiple-instance learning problem. The authors in [4] have also proposed two approaches for maximum margin MIL based on support vector machines. In [5] the authors first calculate a likelihood ratio for each sample in all the bags using a support vector regression (SVR) model. Then they concatenate the likelihood ratios and a binary linear support vector machine (SVM) classifies these vectors as positive and negative. The samples inside each bag are classified using a threshold on their likelihood ratios. One of the works towards convex multiple instance learning was [30], in which the authors propose a model based on matrix completion for MIL. They propose two convex algorithms for matrix completion based on the fixed-point continuation (FPC) method, applied to image categorization.

Action recognition in videos has long been an interesting field of research and different methods have been investigated. Some researches use models that directly utilize bag-of-words (BoWs) representations [31,32]. Some other approaches decompose an action into smaller parts for capturing the local spatial or temporal structure of the activity and to better model the interaction between parts [18,33,11,33]. Some use the global spatio-temporal templates, such as motion history, spatio-temporal shapes and descriptors, the human model changing in time or other templates [34–36]. These approaches try to retain the visual shape and structure of the activity. Recently, most researches use approaches encoding the spatio-temporal layout of a video using a fixed space–time grid. As shown by Laptev et al. [8], compared to simple bag-of words [31], these approaches enhance the recognition rates.

Action localization is usually performed separately from the recognition phase. Most approaches often use spatio-temporal features to localize actions throughout the video. Tran and Yuan [13] use a max-margin structured output regression model. A kernel-based discriminative clustering algorithm is presented in [14], where movie scripts are used as a means of supervision. Tran et al. [15] and Yuan et al. [37] utilize a sub-volume search technique for action localization and [16] proposes a dictionary learning approach for the same problem. Gaidon et al. [17] proposed a model based on a sequence of atomic action units (actoms) which structures the bag-of-words model in a temporal context.

As discussed earlier, joint activity recognition and localization in video sequences could improve performance of both stages. Recently, weakly supervised learning models have been developed for activity recognition/localization and event detection in video sequences [18,20–24,38]. The approach proposed in recognizes activities using a multi-class SVM and an inference is done over temporal segments with dynamic programming. Lan et al. [20] recognize activities from a video while detecting its location and sub-patches to indicate the precise location of the activity. They train a latent SVM with a number of fully annotated videos. They build a model to localize and recognize a single action for each video and infer the sub-patches the action is happening in. The proposed method in [18] extracts spatio-temporal structures by forming clusters of trajectories (termed potential action parts). A graphical model is used to recognize a collection of these clusters as a particular action. Tang et al. [38] use a variable-duration hidden Markov model. Their algorithm divides each video into fixed length temporal segments and builds a structured temporal model on top of the features. The inference process uses an exact MAP inference formulation using dynamic programming, in which they find the sequence of states and durations that maximize a predefined energy function. Their activity segmentation is only restricted to the temporal segments. Another approach is proposed in [22], which constructs a space–time video graph, and finds the sub-graph that maximizes an activity classifier's score. The graph nodes for each video describe local video sub-regions, which are associated with weights learned with a linear SVM to show how much the node supports the action class. The edges between the nodes are determined by proximity in space and time. The detection problem, thus would be equivalent to solving a maximum-weight connected sub-graph problem that could identify both the spatial and temporal regions for a particular activity. Siva and Xiang [21] propose a simple model for solving the activity recognition/localization problem, where a person detector extracts potential action cuboids. The best potential cuboids for each action are selected by a genetic algorithm optimization procedure. Then, a SVM is trained for the recognition purpose. Hoai et al. [24] propose an approach for learning a discriminative sub-window classifier for object/action detection. It simultaneously localizes the instances of the positive class and learns a sub-window SVM to recognize them. In another work, [23] introduces hierarchical space–time segments for action recognition and localization.

Rank Minimization has recently gained a lot of attention, due to the simple, effective success in solving many problems. As denoted by [39,40] the minimization of the rank function can be achieved using the minimizer obtained with the nuclear norm, which is calculated as the sum of singular values. In the field of computer vision, nuclear norm minimization has been applied to many problems, namely: camera calibration [41], structure from motion [42], image segmentation [43], image categorization [44] and many more. In order to solve the rank minimization problem, many approaches are developed, such as Fixed Point Continuation [45], Augmented Lagrangian Multipliers method [46] and Alternating Direction Method [47].

3. Notations

Matrices are characterized by bold capital letters (like $\mathbf{A}$). Non-bold letters are used for scalar variables. The scalar in the row i and column j of matrix $\mathbf{A}$ is denoted by a_{ij}. $\langle \mathbf{a}_1, \mathbf{a}_2 \rangle$ is the inner product between two vectors $\mathbf{a}_1$ and $\mathbf{a}_2$. $\|a\|_2^2 = \langle \mathbf{a}, \mathbf{a} \rangle = \sum_i a_i^2$ indicates the squared Euclidean Norm of the vector $\mathbf{a}$. $\mathrm{tr}(\mathbf{A}) = \sum_i a_{ii}$ is the trace of $\mathbf{A}$. $\|\mathbf{A}\|_F = \sqrt{\mathrm{tr}(\mathbf{A}^\top \mathbf{A})} = \sqrt{\sum_{ij} a_{ij}}$ is the Frobenius Norm of $\mathbf{A}$, and $\|\mathbf{A}\|_*$ the nuclear norm (sum of singular values) of $\mathbf{A}$. $\odot$ designates the Hadamard or elementwise product.

4. Matrix completion for multi-label classification

Matrix Completion is the process of recovering a matrix from a sampling of its entries. We are interested in recovering a data matrix $\mathbf{D}$ from a matrix $\mathbf{D}_0$, in which we only get to observe a number of its entries, which is comparably much smaller than the total number of elements in the matrix. Let Ω denote the set of known entries. With sufficiently large measurements and uniformly distributed entries in the matrix, we can assume that there is only one low-rank matrix with these entries [39]. So the optimization problem would be

$$\begin{aligned} &\text{minimize} \ \ \mathrm{rank}(\mathbf{D}) \\ &\text{subject to} \ \ \mathbf{D}_{ij} = \mathbf{D}_{0_{ij}} \quad (i,j) \in \Omega. \end{aligned} \tag{1}$$

Unfortunately, this is a NP-hard optimization problem and all known algorithms, which provide exact solutions, require exponential time

complexities relative to the matrix dimension, d. But as denoted by [40,39], if a matrix has rank r, then it should have exactly r nonzero singular values. Thus, the rank function could be characterized as the number of non-vanishing singular values. A simple estimate of the rank function can be defined as the sum of the singular values, σ_k, over the constraint set, which is called the nuclear norm:

$$\|\mathbf{D}\|_* = \sum_{k=1}^{d} \sigma_k \|\mathbf{D}\|. \tag{2}$$

An error term is incorporated, for robustness to a level of noise and outliers. The amount of error is controlled by a loss function, $l(.)$. To avoid trivial solutions, this amount of error is also put in the minimization objective along with the matrix rank approximation (nuclear norm):

$$\begin{aligned} &min \|\mathbf{D}\|_* + l(\mathrm{E}) \\ &\text{subject to } \mathbf{D} = \mathbf{D_0} + \mathbf{E} \\ &\quad \mathbf{E}_{\mathrm{ij}} = 0 \ \text{ for } \ (i,j) \notin \Omega. \end{aligned} \tag{3}$$

Recently, this formulation has been used for classification tasks. A classification task is to learn the connection between the space of features $\mathbf{X}$ and the space of labels $\mathbf{Y}$, from N_{tr} training instances. Let m be the number of different classes (the number of labels), n the dimensionality of the feature space, N the number of total instances, and N_{tr} and N_{tst} the number of training and testing instances, respectively. As noted by Goldberg et al. [45] the problem of classifying N_{tst} test entries can be cast as a Matrix Completion task. To this end, we can concatenate all labels and features into a single matrix:

$$\mathbf{D_0} = \begin{bmatrix} \mathbf{Y_{tr}} & \mathbf{Y_{tst}} \\ \mathbf{X_{tr}} & \mathbf{X_{tst}} \\ \mathbf{1}^\top & \end{bmatrix}, \tag{4}$$

where $\mathbf{Y_{tr}} \in \mathbb{R}^{m \times N_{tr}}$ and $\mathbf{Y_{tst}} \in \mathbb{R}^{m \times N_{tst}}$ are the test and training labels and $\mathbf{X_{tr}} \in \mathbb{R}^{n \times N_{tr}}$ and $\mathbf{X_{tst}} \in \mathbb{R}^{n \times N_{tst}}$ are the test and training feature vectors, respectively. If a linear classification model holds, the above matrix should be rank deficient. In this formulation, the classification process would be defined as filling the unknown entries in $\mathbf{Y_{tst}}$ such that the Nuclear Norm of $\mathbf{D_0}$ is minimized. This could be done via a convex minimization process [45,47,30]. In practice, we have errors and incomplete data in the training features and labels. So, we define the set of known entries in $\mathbf{D_0}$ as Ω_X and Ω_Y and zero out unknown entries:

$$\mathbf{D} = \begin{bmatrix} \mathbf{D_Y} \\ \mathbf{D_X} \\ \mathbf{D_1} \end{bmatrix} = \begin{bmatrix} \mathbf{Y_{tr}} & \mathbf{Y_{tst}} \\ \mathbf{X_{tr}} & \mathbf{X_{tst}} \\ 1^\top & \end{bmatrix} + \begin{bmatrix} \mathbf{E_{Y_{tr}}} & \mathbf{0} \\ \mathbf{E_{X_{tr}}} & \mathbf{E_{X_{tst}}} \\ \mathbf{0}^\top & \end{bmatrix}. \tag{5}$$

$\mathbf{D_X}$ and $\mathbf{D_Y}$ stand for the feature and label rows, and $\mathbf{D_1}$ is the last row of the matrix. Therefore, the classification process would be posed as finding the best $\mathbf{Y_{tst}}$ and the error matrix, $\mathbf{E}$ such that the rank of $\mathbf{D} = \mathbf{D_0} + \mathbf{E}$ is minimized [45]. This would be equivalent to [30]:

$$\begin{aligned} &\min_{\mathrm{D}} \|\mathbf{D}\|_* + \frac{1}{|\Omega_X|} \sum_{ij \in \Omega_X} c_x\left(\mathbf{E_{X_{ij}}}\right) + \frac{\lambda_1}{|\Omega_Y|} \sum_{ij \in \Omega_Y} c_y\left(\mathbf{E_{Y_{ij}}}\right) \\ &\text{subject to } \mathbf{D} = \mathbf{D_0} + \mathbf{E}, \mathbf{D_1} = \mathbf{1}^\top, \end{aligned} \tag{6}$$

where $c_y(.)$ is a log loss function to emphasize the error on entries switching classes and $c_x(.)$ is a least squares error. These two terms are to avoid trivial solutions and to penalize large distortions of $\mathbf{D}$. The parameter λ_1 is a positive trade-off weight [45,30]. This minimization problem can be solved using a Fixed Point Continuation (FPC) method [45] or an Alternating Direction Method (ADM) of multipliers [47].

5. Activity recognition and localization

5.1. Video representation

Each video in the dataset is represented with the concatenation on a number of histograms of densely sampled features. We use histogram of gradient (HoG), histogram of optical flow (HoF) [8] and histogram of the oriented edges of the motion boundaries (HoMB) [10] descriptors. As shown by Laptev et al. [8], compared to the bag-of-words approaches, encoding the spatio-temporal layout of a video with a fixed space–time grid enhances the recognition rates. These histograms are computed on a regular grid at three different scales. For each descriptor (HoG, HoF, HoMB) an independent dictionary is used. This is done by using K-means, and quantizing all descriptors to the closest ℓ_2 distance dictionary element. The concatenation of all three histograms forms the video descriptor, $\mathbf{h_i}$, a column vector of size n.

As a result, the feature vector for each video forms a single column in $\mathbf{D_X}$, and the video labels are accommodated in the corresponding column in $\mathbf{D_Y}$. Training and testing samples compose the data matrix. We consider some extra columns in the matrix in which we will recover a representative feature vector for each label (activity class), which will

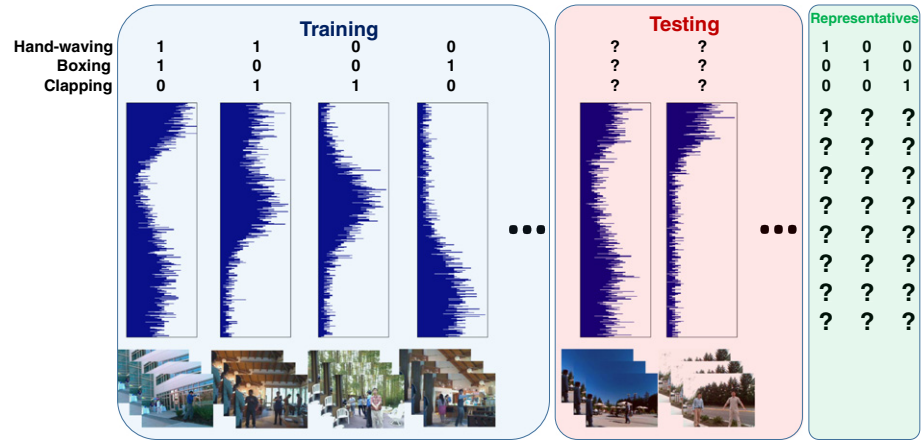


Fig. 3. Illustration of how the matrix $\mathbf{D}$ is composed, using the videos from MSR2 action dataset. Histogram features of each video, alongside its label(s) compose matrix columns. Training, testing and class representatives are concatenated into a matrix. Matrix completion will determine the labels for the test instances, as well as the representative feature vectors for each class.

be discussed in details later. Like discussed in the previous section, Ω_X and Ω_Y denote the known entries in the matrix. The matrix completion process will recover the missing entries in our matrix. For our case, the missing entries are the test labels and the feature vectors representatives of each class. The recovered labels are used to determine the class or classes of activities in each test video, and a search throughout the video would find us the spatio-temporal location of each activity of interest, using its class representative.

5.2. Formulation

We define a matrix $\mathbf{D}$, as follows, to contain both training and testing instances, as well as a set of columns to find representatives for each activity class:

$$\mathbf{D} = \begin{bmatrix} \mathbf{D_Y} \\ \mathbf{D_X} \\ \mathbf{D_1} \end{bmatrix} = \begin{bmatrix} \mathbf{Y_{tr}} & \mathbf{Y_{tst}} & \mathbf{Y_{rep}} \\ \mathbf{X_{tr}} & \mathbf{X_{tst}} & \mathbf{X_{rep}} \\ & \mathbf{1}^\top & \end{bmatrix} + \begin{bmatrix} \mathbf{E_{Y_{tr}}} & \mathbf{0} & \mathbf{0} \\ \mathbf{E_{X_{tr}}} & \mathbf{E_{X_{tst}}} & \mathbf{0} \\ & \mathbf{0}^\top & \end{bmatrix}, \tag{7}$$

where $\mathbf{Y_{rep}}$ would be an $m \times m$ identity matrix. Each column of this matrix would be reserved for one activity class and the recovered corresponding column in $\mathbf{X_{rep}}$ will be a feature vector representative for that class. Therefore, the objective function would be defined as minimizing the rank of the data matrix $\mathbf{D}$, together with the error, matrix $\mathbf{E}$. A sample composition of matrix $\mathbf{D}$ for videos from the multi-label MSR2 action dataset is shown in Fig. 3. Since, we use histograms of quantized features to describe videos, our feature vectors are composed of non-negative values. Thus, we need to ensure that the completion process does not inject negative entries in the matrix. Consequently, we add a constraint $\mathbf{D_X} \geq 0$:

$$\begin{aligned} &\min_{\mathbf{D},\mathbf{X_{rep}}} \|\mathbf{D}\|_* + \frac{\lambda_1}{|\mathbf{D_X}|} \sum_{i,j \in \mathbf{D_X}} l_x\left(d_{ij}, d_{0_{ij}}\right) + \frac{\lambda_2}{|\mathbf{D_Y}|} \sum_{i,j \in \mathbf{D_Y}} l_y\left(d_{ij}, d_{0_{ij}}\right) \\ &\text{subject to } \mathbf{D} = \mathbf{D_0} + \mathbf{E}, \mathbf{D_1} = \mathbf{1}^\top, \mathbf{D_X} \geq 0, \end{aligned} \tag{8}$$

where $l_y(.,.)$ is a log loss function, used to penalize labels changing from one class to the other. $l_x(.,.) = \chi^2(.,.)$ is defined as a Pearson's distance between the entries of the initial and recovered data matrix, which has an asymmetric characteristic.

$$l_x(a,b) = \chi^2(a,b) = \frac{(a-b)^2}{a+b}, \tag{9}$$

$$l_y(a,b) = \frac{1}{\gamma} log\left(1 + e^{(-\gamma(b-a))}\right). \tag{10}$$

A convex minimization process could solve the problem and recover the best matrix. This is explained in details in the next subsection.

5.3. Optimization algorithm

Let us, for simplicity, replace the second and the third terms in the objective function in (8) with $f(\mathbf{E_X})$ and $g(\mathbf{E_Y})$, respectively. Now, we need to convert it to the following equivalent problem:

$$\begin{aligned} &\min_{\mathbf{D},\mathbf{E}} \|\mathbf{D}\|_* + f(\mathbf{E_X}) + g(\mathbf{E_Y}) \\ &\text{subject to } \mathbf{D} = \mathbf{D_0} + \mathbf{E}, \mathbf{D_X} = \mathbf{U}, -\mathbf{U} \leq 0, \end{aligned} \tag{11}$$

The $\mathbf{D_1} = \mathbf{1}^\mathrm{T}$ is enforced by keeping the last row of $\mathbf{E}$ equal to $\mathbf{0}^\top$ [46]. This problem could be solved using an ADM algorithm. The Lagrangian function would be:

$$\begin{aligned} &\|\mathrm{D}\|_* + f(\mathbf{E_X}) + g(\mathbf{E_Y}) + \langle \mathscr{L}_1, \mathbf{D} - \mathbf{D_0} - \mathbf{E} \rangle \\ &\quad + \langle \mathscr{L}_2, \mathbf{D_X} - \mathbf{U} \rangle + \frac{\mu}{2}\left(\|\mathbf{D} - \mathbf{D_0} - \mathbf{E_X}\|_F^2 + \|\mathbf{D_X} - \mathbf{U}\|_F^2\right), \end{aligned} \tag{12}$$

where $\mathscr{L}_1$ and $\mathscr{L}_2$ are the Lagrange multipliers. The variables would be updated iteratively as illustrated in Algorithm 1. To ensure the convergence of the algorithm, we need to satisfy some KKT conditions [48] as well. They are discussed in the next subsection.

Algorithm 1. Non-negative matrix completion, using alternating direction method (ADM) of multipliers.

Input: Initial data matrix $\mathbf{D} = \mathbf{D_0}$, parameters λ_1 and λ_2.
Output: Completed matrix $\mathbf{D}$
$\mathscr{L}_{1_0} = \mathscr{L}_{2_0} = 0, \mu_0 > 0, \rho = 1.1, \mathbf{E_0} = \mathbf{0}, \mathbf{U_0} = \mathbf{0}$.
$k = 0$
while not converged **do**
1. Fix all other variables and update $\mathbf{D_{k+1}}$:
   $\mathbf{D_{k+1}} = \underset{\mathbf{D}}{\operatorname{argmin}} \frac{1}{\mu_k}\|\mathbf{D_k}\|_* + \frac{1}{2}\|\mathbf{D_k} - (\mathbf{D_0} + \mathbf{E_k} - \frac{\mathscr{L}_{1_k}}{\mu_k})\|_F^2$,
2. Fix all other variables and update $\mathbf{E_{X_{k+1}}}$:
   $\mathbf{E_{X_{k+1}}} = \underset{\mathbf{E_X}}{\operatorname{argmin}} \frac{1}{\mu_k} f(\mathbf{E_{X_k}}) + \frac{1}{2}\|\mathbf{E_{X_k}} - (\mathbf{D_{0X}} - \mathbf{D_{X_{k+1}}} + \frac{\mathscr{L}_{1_k}}{\mu_k})\|_F^2$,
3. Fix all other variables and update $\mathbf{E_{Y_{k+1}}}$:
   $\mathbf{E_{Y_{k+1}}} = \underset{\mathbf{E_Y}}{\operatorname{argmin}} \frac{1}{\mu_k} g(\mathbf{E_{Y_k}}) + \frac{1}{2}\|\mathbf{E_{Y_k}} - (\mathbf{D_{0Y}} - \mathbf{D_{Y_{k+1}}} + \frac{\mathscr{L}_{1_k}}{\mu_k})\|_F^2$,
4. Set the $\mathbf{E_k}$ matrix as: $\mathbf{E_k} = \begin{bmatrix} \mathbf{E_{X_k}}^\top & \mathbf{E_{Y_k}}^\top & \mathbf{0} \end{bmatrix}^\top$,
5. Fix all other variables and update $\mathbf{U_{k+1}} = \mathbf{D_{X_{k+1}}} + \frac{\mathscr{L}_{2_k}}{\mu_k}$,
6. Update the multipliers, $\mathscr{L}_1$ and $\mathscr{L}_2$:
   $\mathscr{L}_{1_{k+1}} = \mathscr{L}_{1_k} + \mu_k(\mathbf{D_{k+1}} - \mathbf{D_0} - \mathbf{E_{k+1}})$,
   $\mathscr{L}_{2_{k+1}} = \mathscr{L}_{2_k} + \mu_k(\mathbf{D_{X_{k+1}}} - \mathbf{U_{k+1}})$,
7. Update parameter μ_{k+1} as: $\mu_{k+1} = \min(\rho\mu_k, 10^{10})$.
8. $k = k + 1$
9. Check the convergence condition:
   $(\mathbf{D_k} - \mathbf{D_0} - \mathbf{E_k} \to 0) \wedge (\mathbf{D_{X_k}} - \mathbf{U_k} \to 0) \wedge$
   $(\mathscr{L}_{1_k} \odot \mathbf{U_k} \to 0) \wedge (-\mathbf{U_k} \leq 0) \wedge (\mathscr{L}_{2_k} \geq 0)$.
end while

Step 1 of the algorithm, which consists a nuclear norm minimization could be solved using singular value thresholding (SVT) algorithm [49]. Fortunately, we do not have to solve the problem exactly, and updating $\mathbf{D}$ only once would still satisfy the convergence properties of the problem [46]. So, it is solved as

$$(\mathbf{U}, \mathbf{S}, \mathbf{V}) = \mathrm{svd}\left(\mathbf{D_k} - \mathbf{D_0} - \mathbf{E_k} - \frac{\mathscr{L}_1}{\mu}\right), \tag{13}$$

$$\mathbf{D_{k+1}} = \mathbf{U}\mathscr{S}_\mu^{-}1[\mathbf{S}]\mathbf{V}^\top, \tag{14}$$

where $\mathbf{USV}^\top$ is the singular value decomposition of the matrix, and $\mathscr{S}_\epsilon[.]$ is a shrinkage or a proximal operator for the nuclear norm, defined as

$$\mathscr{S}_\epsilon[x] = \begin{cases} x - \epsilon & \text{if } x > \epsilon, \\ x + \epsilon & \text{if } x < -\epsilon, \\ 0 & otherwise. \end{cases} \tag{15}$$

5.4. Convergence and computational complexity

The convergence properties of the optimization algorithms are dependent to the choice and variations in the parameter μ [46] and KKT (Karush–Kuhn–Tucker) conditions [48,50]. Let's call a combination of the variables present in the optimization process W:

$$W := (\mathrm{D}, \mathrm{E}, \mathrm{U}, \mathscr{L}_1, \mathscr{L}_2). \tag{16}$$

In order for W to be a KKT point, in the KTH iteration of the algorithm, it should satisfy the following conditions, which will pose as the stopping criterion of the algorithm, as well:

$$\mathbf{D_k} - \mathbf{D_0} - \mathbf{E_k} = 0, \tag{17}$$

$$\mathbf{D_{X_k}} - \mathbf{U_k} = 0, \tag{18}$$

$$\mathscr{L}_{1_k} \odot \mathbf{U_k} = 0, \tag{19}$$

$$-\mathbf{U_k} \le 0, \tag{20}$$

$$\mathscr{S}_{2_k} \ge 0. \tag{21}$$

In order to study the convergence properties of the algorithm, we need to first ensure the boundedness of the variables, incorporated in the optimization procedure.

Lemma 5.1. *The sequences* $\{\mathscr{S}_{1_k}\}$ *and* $\{\mathscr{S}_{2_k}\}$ *are bounded.*

Proof. Let's assume $\{\mathbf{U_k^*}\}$, $\{\mathbf{D_k^*}\}$ and $\{\mathbf{E_k^*}\}$ are the optimal values of the above optimization problem. Then:

$$\begin{aligned} 0 &\in \partial_{\mathbf{D}} \mathscr{L}(\mathbf{D_k^*}, \mathbf{E_k^*}, \mathbf{U_k^*}, \mathscr{S}_{1_k}, \mathscr{S}_{2_k}, \mu_k), \\ 0 &\in \partial_{\mathbf{E}} \mathscr{L}(\mathbf{D_k^*}, \mathbf{U_k^*}, \mathbf{E_k^*}, \mathscr{S}_{1_k}, \mathscr{S}_{2_k}, \mu_k), \end{aligned} \tag{22}$$

Therefore,

$$\{\mathscr{S}_{1_k}, \mathscr{S}_{2_k}\} \in \partial_{\mathbf{D}} \|\mathbf{D_k^*}\|_*, \quad \{\mathscr{S}_{1_k}, \mathscr{S}_{2_k}\} \in \partial_{\mathbf{E}} (f(\mathbf{E_{X_k}}) + g(\mathbf{E_{Y_k}})). \tag{23}$$

Consequently, $\{\mathscr{S}_{1_k}\}$ and $\{\mathscr{S}_{2_k}\}$ are bounded. □

Lemma 5.2. *If* μ_k *satisfies the condition* $\sum_{k=1}^{+\infty} \mu_k^{-2} \mu_{k+1} < +\infty$, *the sequences* $\{\mathbf{E_k}\}$, $\{\mathbf{D_k}\}$, $\{\mathbf{U_k}\}$, $\{\mathbf{U_k^*}\}$, $\{\mathbf{D_k^*}\}$ *and* $\{\mathbf{E_k^*}\}$ *are bounded.* ($\{\mathbf{U_k^*}\}$, $\{\mathbf{D_k^*}\}$ *and* $\{\mathbf{E_k^*}\}$ *are the optimal values of the optimization problem*).

Proof. As far as we are solving a minimization problem, we can say:

$$\begin{aligned} &\mathscr{L}(\mathbf{D_{k+1}^*}, \mathbf{E_{k+1}^*}, \mathbf{U_{k+1}^*}, \mathscr{S}_{1_k}, \mathscr{S}_{2_k}, \mu_k) \le \mathscr{L}(\mathbf{D_{k+1}^*}, \mathbf{E_k^*}, \mathbf{U_k^*}, \mathscr{S}_{1_k}, \mathscr{S}_{2_k}, \mu_k) \\ &\le \mathscr{L}(\mathbf{D_k^*}, \mathbf{E_k^*}, \mathbf{U_k^*}, \mathscr{S}_{1_k}, \mathscr{S}_{2_k}, \mu_k) = \mathscr{L}(\mathbf{D_k^*}, \mathbf{E_k^*}, \mathbf{U_k^*}, \mathscr{S}_{1_{k-1}}, \mathscr{S}_{2_{k-1}}, \mu_{k-1}) \\ &+ \frac{1}{2}\mu_{k-1}^{-2}(\mu_{k-1} + \mu_k)\left(\|\mathscr{S}_{1_k} - \mathscr{S}_{1_{k-1}}\|_F^2 + \|\mathscr{S}_{2_k} - \mathscr{S}_{2_{k-1}}\|_F^2\right). \end{aligned} \tag{24}$$

and with the boundedness of $\{\mathscr{S}_{1_k}\}$ and $\{\mathscr{S}_{2_k}\}$, and $\sum_{k=1}^{+\infty} \mu_k^{-2} \mu_{k+1} < +\infty$ we can say that $\{\mathscr{L}(\mathbf{D_{k+1}^*}, \mathbf{E_{k+1}^*}, \mathbf{U_{k+1}^*}, \mathscr{S}_{1_k}, \mathscr{S}_{2_k}, \mu_k)\}$ is bounded and as a result $\{\mathbf{U_k^*}\}$, $\{\mathbf{D_k^*}\}$ and $\{\mathbf{E_k^*}\}$ are bounded, as well. The boundedness of $\{\mathbf{U_k}\}$, $\{\mathbf{E_k}\}$ and $\{\mathbf{D_k}\}$ is also proved in a same way. □

Theorem 5.3. *If the sequences of Lagrangian multipliers are bounded, then for the sequence* W_k *we can say:* $\sum_{k=0}^{\infty} \|\mathscr{S}_{1_{k+1}} - \mathscr{S}_{1_k}\|_F^2 < \infty$ $\sum_{k=0}^{\infty} \|\mathscr{S}_{2_{k+1}} - \mathscr{S}_{2_k}\|_F^2 < \infty$ *Then*

$$\lim_{k \to +\infty} (W_{k+1} - W_k) = 0, \tag{25}$$

and each W_k *point is a KKT point and is a solution in the problem.*

Proof. With regard to the above two lemmas, the conditions of this theorem would be satisfied. Therefore, it could be proved exactly the same as described in [50]. This theorem guarantees that the achieved solution is feasible. □

Theorem 5.4. *For the above algorithm* ($\mathbf{D_k^*}$, $\mathbf{E_k^*}$) *is an optimal solution, and the convergence rate is of at least* $O(\mu_k^{-1})$.

$$\left|\|\mathbf{D_k}\|_* + f(\mathbf{E_{X_k}}) + g(\mathbf{E_{Y_k}}) - F^*\right| = O(\mu_k^{-1}) \tag{26}$$

where F^* *is the optimal solution of the problem in (12).*

Proof. Since F^* is the optimal solution:

$$\begin{aligned} &\mathscr{L}(\mathbf{D_{k+1}^*}, \mathbf{E_{k+1}^*}, \mathbf{U_{k+1}^*}, \mathscr{S}_{1_k}, \mathscr{S}_{2_k}, \mu_k) = \min_{\mathbf{D},\mathbf{E},\mathbf{U}} \mathscr{L}(\mathbf{D}, \mathbf{E}, \mathbf{U}, \mathscr{S}_{1_k}, \mathscr{S}_{2_k}, \mu_k) \\ &\le \min_{\mathbf{D}+\mathbf{E}=\mathbf{D_0}, \mathbf{U}=\mathbf{D_0}} \mathscr{L}(\mathbf{D}, \mathbf{E}, \mathbf{U}, \mathscr{S}_{1_k}, \mathscr{S}_{2_k}, \mu_k) = F^*, \end{aligned} \tag{27}$$

so

$$\begin{aligned} &\|\mathbf{D_{k+1}^*}\|_* + f\left(\mathbf{E_{X_{k+1}}^*}\right) + g\left(\mathbf{E_{Y_{k+1}}^*}\right) \\ &= \mathscr{L}(\mathbf{D_{k+1}^*}, \mathbf{E_{k+1}^*}, \mathbf{U_{k+1}^*}, \mathscr{S}_{1_k}, \mathscr{S}_{2_k}, \mu_k) - \frac{1}{2\mu_k}\left(\|\mathscr{S}_{1_k} - \mathscr{S}_{1_{k-1}}\|_F^2 + \|\mathscr{S}_{2_k} - \mathscr{S}_{2_{k-1}}\|_F^2\right) \\ &\le F^* - \frac{1}{2\mu}\left(\|\mathscr{S}_{1_k} - \mathscr{S}_{1_{k-1}}\|_F^2 + \|\mathscr{S}_{2_k} - \mathscr{S}_{2_{k-1}}\|_F^2\right) = F^* - O(\mu_k^{-1}). \end{aligned} \tag{28}$$

Due to the boundedness of all the variables, the theorem is proved. □

We can conclude that if μ_k is increasing slightly in each iteration, the algorithm is Q-linearly convergent. If μ_k will be increasing faster, the algorithm would even be converged faster, but increasing this variable too much would cause a slower convergence in the singular value thresholding sub-problem. As a result, a good choice for the sequence of $\{\mu_k\}$ variables yields in a decrease in the number of required SVD operations.

The computational complexity of the algorithm is very easy to compute, since the algorithm is a single straightforward loop. The most computationally expensive stage in the loop is to compute an SVD of a matrix with $(m + n) \times N$ elements. Since $m + n < N$, we can say that the complexity of the SVD part is $O(N^3)$. Therefore, the computational complexity of the whole algorithm would be $O(qN^3)$, where q is the number of iterations (in practice $q << N$). As discussed above, q is dependent to the choice of $\{\mu_k\}$ and the Q-linear convergence rate.

5.5. Action localization

Let's say each video i is represented with a histogram $\mathbf{h_i} = [b_1, b_2, \ldots, b_n]$. In general, histogram of videos is a linear combination of histograms of all its sub-regions. For instance, if there are two activities, a_1 and a_2 in the video, the histogram of the whole video would be $\mathbf{h_i} = \alpha \mathbf{h_{a_1}} + \beta \mathbf{h_{a_2}} + \gamma \mathbf{h_{bg_i}}$, where $\mathbf{h_{a_1}}$ is the histogram for the first activity, $\mathbf{h_{a_2}}$ is the histogram for the second activity and $\mathbf{h_{bg_i}}$ is the additive noise from the background and clutter present in the video i. This simple characteristic of the histograms (like also used in [44]) enables us extract video sub-region histograms, with our matrix completion framework. With respect to this characteristic of the histograms, minimizing the rank of the data matrix (shown in Fig. 3) would decompose each activity histogram and would form the entries in $\mathbf{X_{rep}}$.

When minimizing the rank of the matrix, we are actually maximizing the linear relations between both the rows (corresponding to row-rank) and the columns (corresponding to column-rank). We can say that the row-rank minimization helps inferring the label(s) for each video and the column-rank minimization contributes to finding the

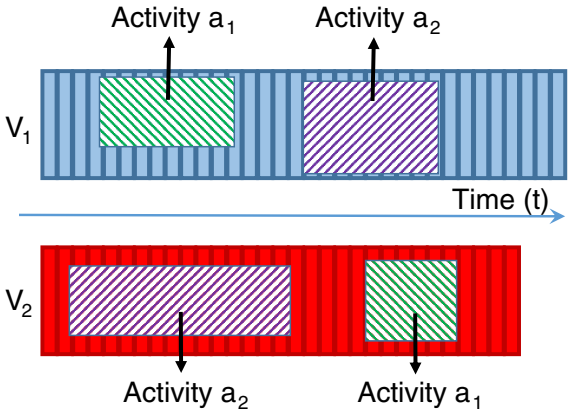


Fig. 4. Two videos, V_1 and V_2, both containing two activities, a_1 and a_2. In V_1, first activity a_1 happens and then a_2, and in V_2 they happen in a reverse order. Our method could recognize and localize both activities (see the text).

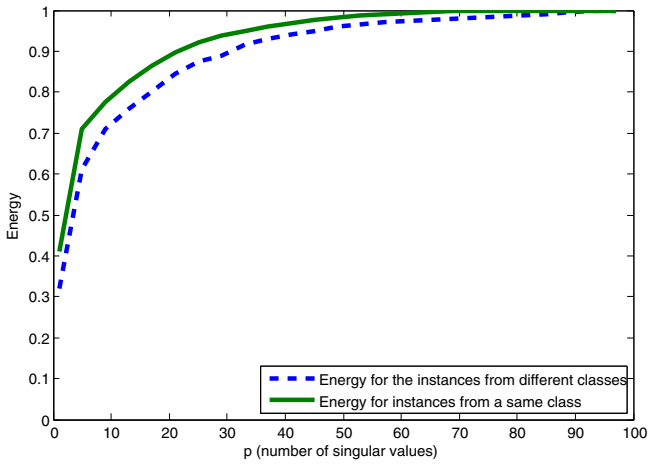


Fig. 5. The mean energy signals for $\mathscr{X}_i$s and $\mathscr{Y}_j$s as a function of p. The first singular values have more energy for the case of matrices with all columns of same class instances, compared to the case that the matrix is composed of features from different classes.

representative feature vectors to equip the action localization process. Each column contains the histogram of a single video. So, in order for the column-rank to be minimized, the process should better correlate the unknown values in the representative feature vectors to the other columns. Since these columns hold values for the corresponding labels, the unknown feature vector values of these columns will be a compromise from all known columns containing the same labels.

As an example, consider the case illustrated in Fig. 4. In this figure, there are two videos (V_1 and V_2), both containing two a_1 and a_2 actions in different orders. These two videos together with all other videos in the dataset are put in the matrix (Fig. 3). All these videos contribute in creating representative histogram feature vectors for each activity class. Let's call the histogram feature vectors of V_1 and V_2 as $\mathbf{h_1}$ and $\mathbf{h_2}$. Therefore, we could say $\mathbf{h_1} = \alpha_1\mathbf{h}_{\mathbf{a}_1} + \beta_1\mathbf{h}_{\mathbf{a}_2} + \gamma_1\mathbf{h}_{\mathbf{bg}_1}$ and $\mathbf{h_2} = \alpha_2\mathbf{h}_{\mathbf{a}_1} + \beta_2\mathbf{h}_{\mathbf{a}_2} + \gamma_2\mathbf{h}_{\mathbf{bg}_2}$. When the matrix is completed, ideally $\mathbf{h}_{\mathbf{bg}_1}$ and $\mathbf{h}_{\mathbf{bg}_2}$ will be removed from the data and be accommodated in the error matrix, $\mathbf{E}$. Furthermore, $\mathbf{h}_{\mathbf{a}_1}$ and $\mathbf{h}_{\mathbf{a}_2}$ will contribute in building the representative histogram feature vectors for their corresponding activity classes. We use the representative histogram feature vectors for a_1 and a_2 activities, and search the histograms to find the best spatio-temporal segment with the minimum distance with these representatives. It is very important to note that the proposed method is a transductive method (like [45]), which means that all samples shall be put together in the matrix and the decisions on the labels of the test videos are acquired all at once. The same applies for when we are looking for representative feature vectors. When there are quite enough number of videos (columns in the matrix), we can expect getting good representative feature vectors.

For each video labeled with an activity, our task would be to search throughout the video and find the spatio-temporal region, which best matches the representative histogram of that specific activity class. One naive solution for the search procedure could be a cuboid search, which is quite similar to sliding window search in the object recognition context. But cuboid search would be computationally quite expensive. Therefore, we first extract potential activity regions in the video and only search among these regions.

In order to extract the potential activity regions, we use person detectors, face detectors and space–time interest point detectors on the video. These detectors are used to help reduce the search space, in order not to perform a full cuboid search on the videos. On each video a pre-trained person detector [51] is run on every $\mathscr{F}^{th}$ frames and a bounding box with a margin is created around that. We track this bounding box with a Meanshift tracker [52] throughout the video and omit repeated results. We also run a face detector on the same $\mathscr{F}^{th}$ frames of the video, and create same type of bounding boxes around them. We further detect all the space–time interest points (STIP) [9], in the video. This detector is based on the extension of Harris operator to both space and time. We name these three detectors as $\mathbf{P_i}$, $\mathbf{F_i}$ and $\mathbf{IP_i}$, for each video V_i. The first two ones, detect the presence of a human, and the latter one detect segments with lots of motion (possibly containing objects and humans). In order to model interactions between humans and objects, we further merge co-occurring segments. Co-occurring segments are those happening in at least 50% temporal duration, together. Algorithm 2 shows the procedure for extracting potential spatio-temporal segments, $\mathbf{S_i}$ for each video V_i. Now, all the $\mathbf{S_i}$s are converted into their respective histograms of densely sampled quantized features.

Algorithm 2. Algorithm for extracting potential spatio-temporal segments.

Input: Video V_i, parameters τ_1 and τ_2.
Output: Potential spatio-temporal segments, $\mathbf{S_i}$.
1. Run a person detector on the video and save all potential regions as a set $\mathbf{P_i}$.
2. Run a face detector on the video and save all potential regions as a set $\mathbf{F_i}$.
3. Detect Space-Time Interest Points (STIP) on the video, generate all the cuboids of at least τ_1 STIPs. Save all these potential regions as the set $\mathbf{IP_i}$.
4. Merge all possible co-occurring regions in $\mathbf{P_i}$ and append them to the set $\mathbf{P_i}$. Do the same with all the segments in $\mathbf{F_i}$ and $\mathbf{IP_i}$, as well.
5. Put all the regions from $\mathbf{P_i}$, $\mathbf{F_i}$ and $\mathbf{IP_i}$, together with all the merged versions of the co-occurring regions in a set $\mathbf{S_i}$.
6. Remove all regions with more than 90% overlap. Only keep one of these regions in $\mathbf{S_i}$.
7. For each region in $\mathbf{S_i}$, calculate a score (Person Detector Score + Face Detector Score + STIP Density Score). Only keep the top τ_2 regions in $\mathbf{S_i}$.

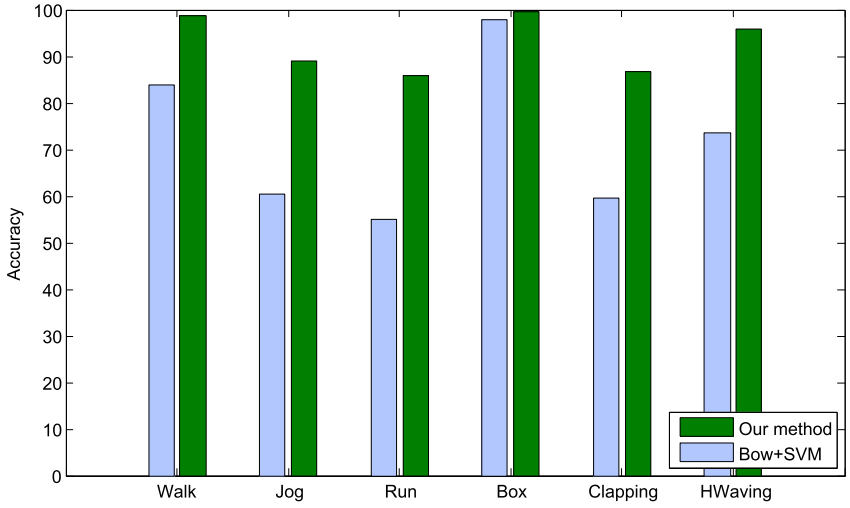


Fig. 6. Per-class classification accuracy for KTH dataset.

Table 1
Recognition accuracy results on KTH dataset. The supervision columns show if the annotation bounding boxes or silhouettes were used for training the models (Full), or not (Weak). Note that our method performs both recognition and localization simultaneously.

Method	Supervision	Accuracy
Laptev et al. [8]	Full	91.8%
Ryoo et al. [25]	Full	91.1%–93.8%
Savarese et al. [55]	Full	86.8%
Niebles et al. [56]	Full	81.5%
Fathi et al. [57]	Full	90.5%
Schuldt et al. [31]	Full	71.7%
Best of Wang et al. [9]	Full	92.1%
Liu et al. [58]	Full	93.8%
Kovashka and Grauman [12]	Full	94.53%
Le et al. [59]	Full	91.4%–93.9%
Our method	Weak	92.6%

Table 2
Recognition accuracy results on Weizmann dataset. Note that our method, unlike others, performs both recognition and localization simultaneously.

Method	Supervision	Accuracy
Zhang et al. [60]	Full	92.89%
Bregonzio et al. [61]	Full	96.66%
Fathi et al. [57]	Full	99.9%
Niebles et al. [56]	Full	90.0%
Hoai et al. [19]	Full	87.7%
Tian et al. [11]	Full	100%
Our method	Weak	98.9%

A simple Euclidean distance is used to determine the best activity candidate for each video. For a test video i, classified as class k, the corresponding representative histogram for that class in $\mathbf{X_{rep}}$ is matched to all the $\mathbf{S_i}$s found in video i and the one with the least distance is selected as the spatio-temporal segmentation for that activity in video i.

6. Experiments

To evaluate the proposed technique for activity recognition, we set up several experiments on various datasets. In this section we review the results and compare them with the state-of-the-art methods.

6.1. Datasets

Several popular action and activity recognition datasets are used for evaluation of the proposed algorithm. We divide the datasets into two board categories: (1) Easy Datasets and (2) Hard Datasets. The former includes KTH [31] and Weizmann [53], which do not have camera motion or occlusion. The latter includes and MSR2 [28], HOHA [8] and UCF Sports [26] action datasets, which contain arbitrary camera motion with partial occlusion during the action. In this second category of the datasets, many actions are defined as interactions between different subjects or with objects in the scene.

KTH action recognition dataset has six types of human actions (walking, jogging, running, boxing, hand waving and hand clapping) performed by 25 subjects, in 2391 video sequences. This dataset is taken with a static camera with 25 fps and a resolution of 160 × 120. *Weizmann* action recognition dataset contains 90 video sequences of resolution 180 × 144 with 50 deinterlaced fps. The dataset has nine different people performing 10 actions (run, walk, skip, jumping-jack, jump-forward-on-two-legs, jump-in-place-on-two-legs, gallop-sideways, wave-two-hands, wave-one-hand and bend).

MSR2 action dataset contains 54 videos and three action categories: boxing, clapping and hand-waving. In this dataset, the videos contain multiple actions and even some have actions occurring at the same time. *HOHA* (Hollywood Human Action) dataset contains 430 videos with a resolution of 240 × 450 and 24 fps. In this dataset, each video sequence contains nuisances such as significant camera motion, rapid scene changes and occasionally significant clutter. Furthermore, actions in this dataset are performed in many different positions and conditions. Furthermore, many actions are defined by the interactions between the subjects and objects. As a result, this dataset is a really challenging one. UCF Sports dataset consists of 150 videos extracted from sports broadcasts. This dataset has more constrained environments, but still there are camera motions, as well as many different lighting and capturing conditions. It is also a challenging dataset due to the large displacements in most of its actions, the cluttered background, and the large intra-class variability.

6.2. Feature vectors

As discussed, we use HoG, HoF and HoMB histogram feature vectors to represent each video. To generate these histograms, dense sampling of features is used, which extracts video space–time blocks from five dimensions (x, y, t, σ, τ). σ and τ are the spatial and temporal scales, respectively. In our experiments, the size of a space–time patch is considered to be 18 × 18 × 10 with 50% overlap between consecutive patches, as also used in [9].

We have adapted a low-rank framework for activity recognition. So, first we need to see if our histogram features are a good fit for this purpose. The feature vectors of instances for same classes should compose lower-rank matrices, than those composed from instances from

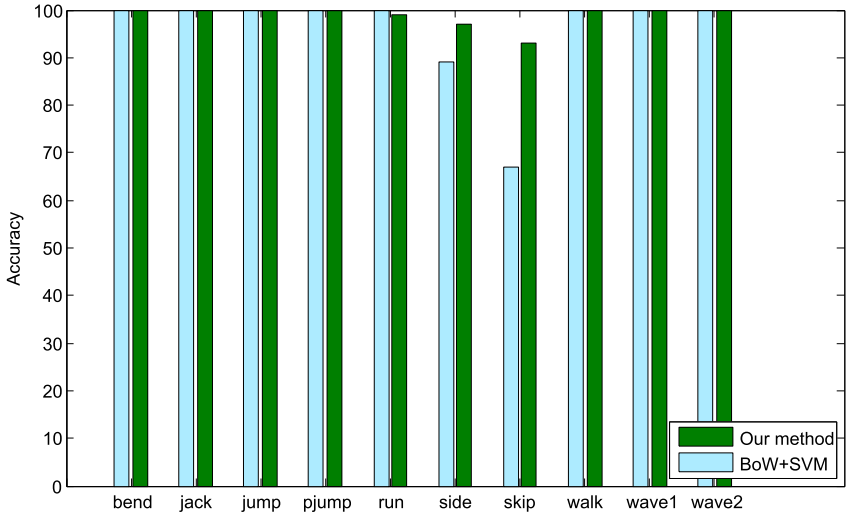


Fig. 7. Per-class classification accuracy for Weizmann dataset.

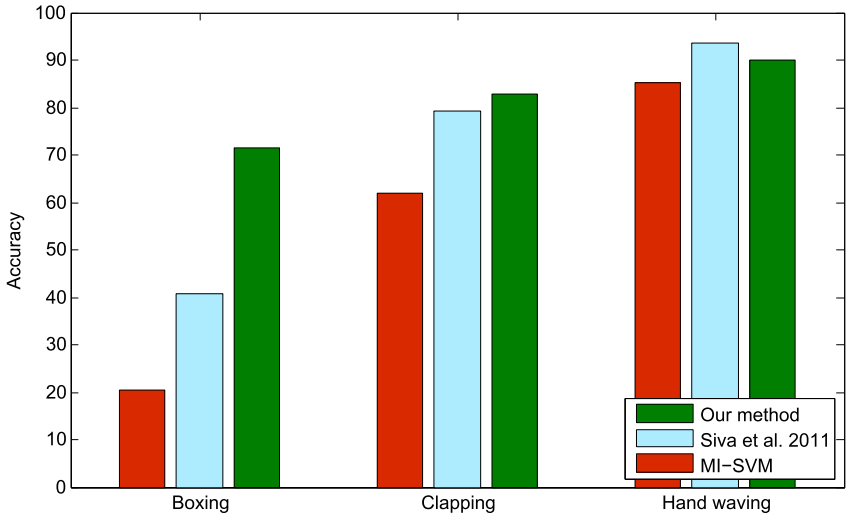


Fig. 8. Per-class classification accuracy for MSR2 dataset.

different classes. To perform such a test, we extract the histogram features (HoG, HoF, HoMB) from the action bounding boxes of the UCF Sports dataset (provided by [26]). For each class i, we construct a matrix $\mathcal{X}_i$ by concatenating all the feature vectors of the same activity class. We also create some matrices $\mathcal{Y}_j$, constructed using activities from two or more classes.

The rank function is modeled by the number of non-zero singular values, and approximated by the sum of the singular values. Therefore, minimizing the rank is equivalent to minimizing the number of non-vanishing singular values. Thus, a matrix with a lower rank would have more energy of the singular values concentrated on its first singular values. We can define an energy function as a function of singular values σ_j [44]:

$$E(X,p) = \frac{\sum_{j=1}^{p}\sigma_j}{\sum_{j=1}^{min(n,D)}\sigma_j}, \tag{29}$$

where p is the number of singular values we want to calculate the energy for, n is the length of the feature vectors and D is the total number of instances in the matrix X. Fig. 5 shows the mean energy signal for $\mathcal{X}_i$s and $\mathcal{Y}_j$s, as described above. As it is obvious, the first singular values have more energy for the case of $\mathcal{X}$ matrices with all columns of same class instances, compared to the case that the matrix is composed of features from different classes.

Table 3
Recognition accuracy results on MSR2 dataset. This dataset is a multi-label dataset, in which there are videos that contain more than one action. Our approach, unlike others, does not require splitting the videos manually to have a single action each.

Method	Experiment conditions	Supervision	Accu.
Tian et al. [62]	Cross dataset (Training: KTH, Testing: MSR2)	Full	78.8%
Yuan et al. [63]	Videos are split to have a single action	Full	58.3%
Siva et al. [21]	Videos are split to have a single action	Weak	71.2%
Our method	Multi-label, simultaneous recognition/localization	Weak	81.5%

6.3. Recognition

Experimental settings for each of the datasets are explained separately. This is because we have adapted similar set of settings to the approaches we compare to. We compare our model with the state-of-the-art models reported in the literature and with the well-known bag-of-words model with SVM classifier with a χ^2 kernel. The results reported here are independent from the localization, which is discussed in the next subsection. Note that, since we use a k-means clustering with

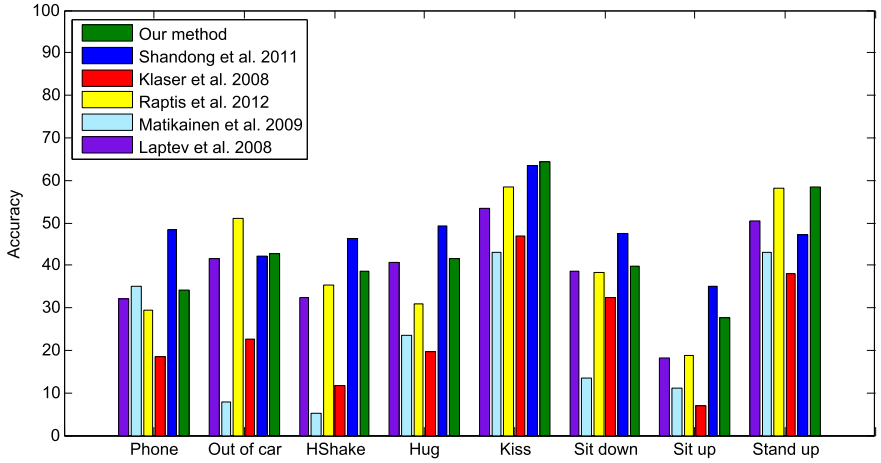


Fig. 9. Per-class classification accuracy for HOHA dataset.

Table 4
Recognition accuracy results on HOHA dataset. Our method, unlike others, is working under weakly supervised settings.

Method	Supervision	Accuracy
Raptis et al. [18]	Full	40.1%
Klaeser et al. [64]	Full	24.7%
Shandong et al. [32]	Full	47.6%
Laptev et al. [8]	Full	38.4%
Matikainen et al. [65]	Full	22.8%
Our method	Weak	43.41%

random initialization for all our models to make the dictionary of the histograms (bags), a deviation of $\pm 0.5\%$ is expected.

6.3.1. KTH dataset

All sequences are divided with respect to the subjects into a training set (16 persons), and a test set (9 persons). Fig. 6 shows the classification accuracy of our model for each of the activity classes in comparisons with the baseline method (SVM + BoW). Table 1 also shows the accuracy results in comparisons with some state-of-the-art methods in the same database. As could be seen, our method gives competitive results with the best results previously reported. Note that our method, both recognizes and localizes the activities in space and time, while others use the annotated activity locations (bounding boxes or silhouettes), for training. This is indicated in the supervision column in the table.

6.3.2. Weizmann dataset

For this dataset we use a leave-one-out cross-validation (LOOCV) method for training and testing the data. Fig. 7 shows the per-class recognition results, compared to the BoW + SVM model. Table 2 illustrates the accuracy results in comparisons with some state-of-the-art methods on Weizmann database.

6.3.3. MSR2 dataset

For the experiments on this dataset, a two-to-one random division of all videos in the dataset is used as the training and testing set. This dataset contains videos with multiple actions happening in the video and in some cases the actions are being performed at the same time. It is a good test for the multi-label settings of our method. Some of the videos in this dataset contain several instances of all activities. Since we try to find each activity in the video sequences and match the first potential region, we split the videos such that each video contains only one instance of each activity class, while there might be several activities from different classes, in each video. Fig. 8 shows the per class accuracy results of our method, compared to the MI-SVM model [4], used in [21]. The MI-SVM model uses a latent SVM framework with a BoW-like set of features. Table 3 shows accuracy results for the recognition task, compared with several state-of-the-art methods. Some methods,

Table 5
Recognition accuracy results on UCF Sports dataset. Some approaches use LOOCV on this dataset, which generally achieve better results, thanks to the easier dataset split for training/testing (upper part). Some other approaches use 103:47 dataset split (lower part). Our method, besides not using the annotations, is competitive even with the LOOCV methods.

Method	Supervision	Accuracy
Kovashka and Grauman [12]	Labels + Annotations	87.3%
Le et al. [59]	Labels + Annotations	86.5%
Wang et al. [9]	Labels + Annotations	85.6%
Wang et al. [10]	Labels + Annotations	88.2%
Wang et al. [33]	Labels + Annotations	89.1%
Tian et al. [11]	Labels + Annotations	75.2%
Ma et al. [23]	Labels	81.7%
Raptis et al. [18]	Labels + Annotations	79.4%
Lan et al. [20]	Labels + Annotations	73.1%
Our method	Labels	85.3%

reported recognition results with cross dataset tests. They train on the KTH dataset and test on MSR2. This is indicated in the table, as well, in the 'experiments condition' column.

6.3.4. HOHA dataset

In this experiment the test set has 211 videos with 217 labels and the training set has 219 videos with 231 labels, all manually annotated [18]. Fig. 9 shows the per-class accuracy results for this dataset. Since this dataset is a very hard task for activity detection, due to the large amount clutter and random motion in the camera, the recognition results on all methods are not very good. But our approach is comparable with all other state-of-the-art methods that are designed specifically for this dataset and such videos. Unlike many other methods, aside from recognition, we also perform the localization task. Table 4 shows the overall accuracy results compared to some other recent methods, on this dataset.

6.3.5. UCF Sports dataset

For this dataset we use a split into 103 trainings and 47 test samples, these are the settings used in [18,20]. This separation minimizes the strong correlation of background cues between the test and training set [18]. Fig. 10 gives the per class classification accuracy for this dataset. As could be seen our method outperforms the BoW + SVM model in almost all classes. Some results reported in the previous works on this dataset use LOOCV (leave-one-out-cross-validation) method, which may take into account the similarity of the background instead of the activity itself. This is because in this sports dataset the background is very similar for sports of a same kind and it can affect the activity recognition rates. Therefore, these methods generally achieve better recognition accuracies. In order to have a better comparative study, we compare our results to both LOOCV and 103:47 dataset split methods. Table 5 shows these recognition rates. The upper part of the table reports

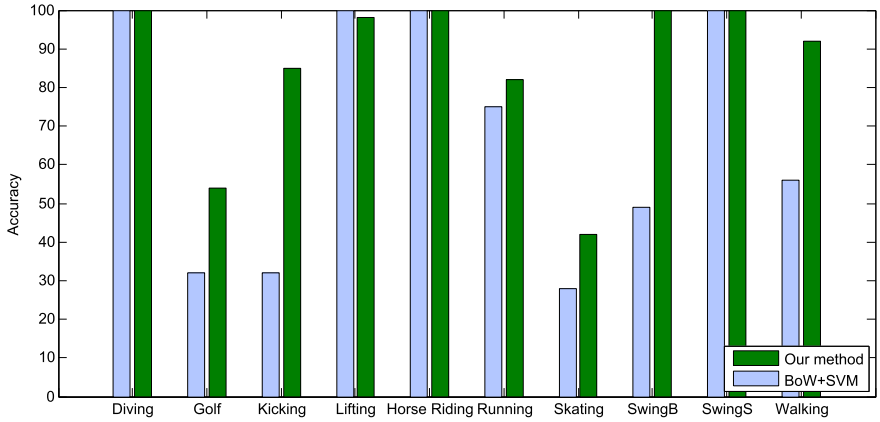
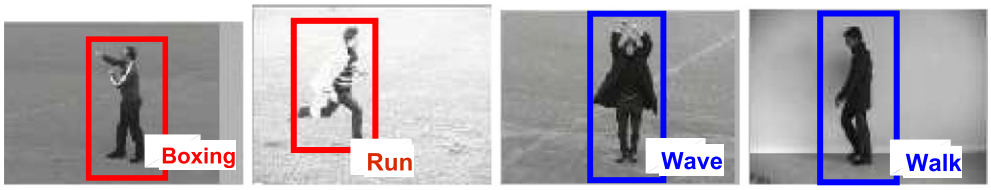
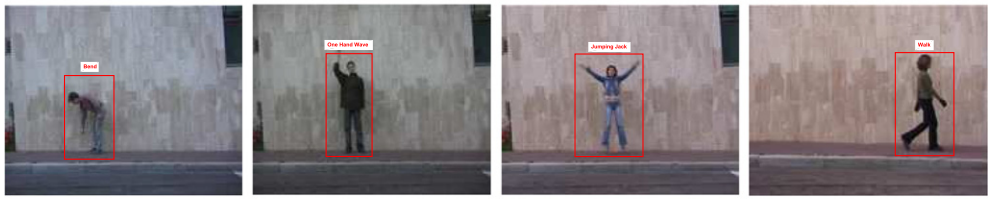


Fig. 10. Per-class classification accuracy for UCF Sports dataset.



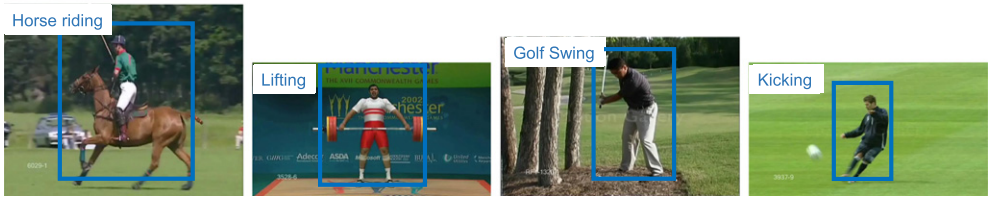
(a) KTH Dataset



(b) Weizmann Dataset

(c) MSR Action Dataset 2

(d) HOHA Dataset



(e) UCF Sports Dataset

Fig. 11. Samples of localization results on different datasets.

results of methods that use LOOCV, and the lower part use an experimental setting, similar to ours. As it is obvious, our results are quite competitive with those fully supervised methods. Fully supervised methods use both the video labels and the bounding box annotations for training their systems. Furthermore, our results are also competitive with those methods that use LOOCV. Although we have a harder dataset split, we achieve comparable results. This could be because we have adapted a so-called transductive method for action recognition.

6.4. Localization

The second functionality of our method is the localization of the activity in the video sequence. As discussed in Section 5.5 the class representatives ($\mathbf{X}_{\mathbf{rep}}$ in (7)) are used to detect the corresponding activity in the video. A search among the potential regions of this video will give us the region which best matches the representative histogram, and is selected as the spatio-temporal segmentation of the activity. In order to evaluate the action localization process, we employ a localization

Table 6
Average Precision (AP) for the actions in the KTH dataset. N/A means not applicable.

Method	Walking	Jogging	Running	Boxing	Hand-wave	Hand-clap	Avg.
Kumar and Patras [66]	N/A	N/A	N/A	0.59	0.97	0.84	0.80
Leibe et al. [67]	N/A	N/A	N/A	0.40	0.95	0.85	0.73
Kumar and Patras [16]	N/A	N/A	N/A	0.68	0.97	0.88	0.84
Our method	0.92	0.81	0.75	0.95	0.81	0.93	0.86

Table 7
Average Precision (AP) of action localization on the MSR2 dataset.

Method	Supervision	Clapping	Boxing	Handwaving
Siva et al. [21]	Full	0.602	0.694	0.700
Siva et al. [21]	Weak	0.326	0.658	0.799
Our method	Weak	0.555	0.701	0.779

score threshold, θ. The score of a detected spatio-temporal segment is defined by averaging the overlap percentage between the segment and the annotation bounding box from the dataset, as defined in [18]. In our experiments, if the detected spatio-temporal segment for an activity has a $\theta = \frac{1}{8}$ overlap with the annotation ground truth from the dataset, the recognition and localization is considered as correct. Some results could be found in Fig. 11.

In order to perform a comparative study between the proposed method and state-of-the-art action localization techniques, we employ the well-known intersection-over-union (IOU) criterion, the average precision (AP) for action localization and the localization score (θ). We also plot precision–recall curve for action localization comparisons. Our results are compared with previous methods, with these metrics, where available. Tables 6, 7 and 8 show comparisons of the results on KTH, MSR2 and UCF Sports datasets, respectively. Note that Lan et al. [20] only provides localization results on a subset of frames on the UCF Sport dataset. Thus, we include results on this subset for comparison, as well. Fig. 12 illustrates the average localization score for different actions in the HOHA dataset, as function of θ, in comparison with [18]. Furthermore, the precision–recall curves for two 'Diving' and 'Hours Riding' actions of the UCF Sports dataset are depicted in Fig. 13. These curves are drawn for the subset of frames used in [20], for comparisons purposes. As could be seen, our method performs quite competitive even to the methods which were designed only for activity localization.

Since the recognition and the localization functions are two separate phases, there is the probability that the selected segment as the best match to the activity is not the correct activity region we were looking for. Therefore, we also take into account the second, third, forth and the fifth regions, and calculate the recognition/localization accuracy for each. Each of these five is called a rank. When we calculate the accuracy for a rank r, if the region of interest is found in the set $1, \ldots, r$ we will consider it as correct, otherwise incorrect. With this definition, we can draw a cumulative match score (CMS) diagram as shown in Fig. 14. The scores in the diagram are the recognition/localization accuracy as a percentage calculated overall in each dataset, independently.

6.5. Discussions

As mentioned before, we have developed a multi-label classification and localization framework for human action recognition. The results on the MSR2 action dataset prove that the multi-label setting is working fine. Moreover, in this paper we proposed an approach that not only recognizes the video scenes to have an activity of interest, it segments the activity spatio-temporally. This is of a great interest, since not much previous methods tackle the problem of segmenting in both time and space. An important note on the recognition results, independent from the segmentation phase, is that it is quite competitive even with those, which do not perform any localization of the activity. This is because when using the whole frame or video features for the recognition, one incorporates many outliers and noise. But in our method we try to take out the noise and use the remainder for the localization task. Taking out the noise helps a better modeling of the data and therefore a better recognition rate.

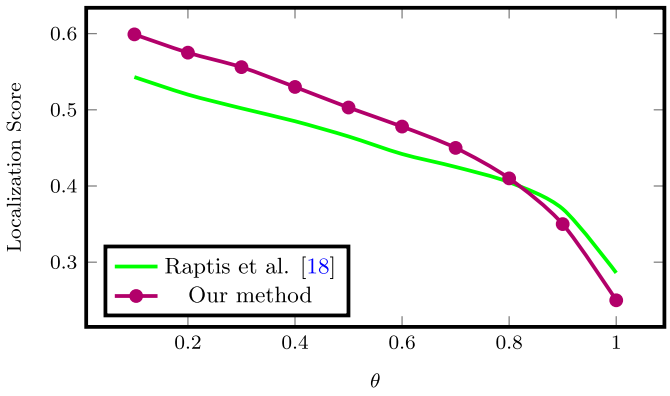


Fig. 12. Average localization scores comparisons between method presented in [18] and our method, as a function of the overlap threshold (θ), for the HOHA dataset. These values for the highest ($\theta = 1$) and the lowest ($\theta = 0.1$) thresholds are 0.599 and 0.25 (our method), 0.543 and 0.286 (Raptis et al. [18]), respectively.

Furthermore, our method works fine for many types of datasets. As explained in the experiments section, we have divided our test cases to Easy and Hard, and reported results on both sets of data. Our method does not put any restrictions on the data, from clutter or noise to camera motion.

7. Conclusion

In this paper, we have developed a multi-label classification framework with a convex optimization process for the problem of activity detection. Our approach uses a non-negative matrix completion framework to recover the labels for the test videos. Moreover, we use the histograms of densely sampled features throughout the video. In the matrix completion procedure, we correct the histograms such that for

Table 8
Localization average IOU Percentage on UCF Sports dataset. N/A means not applicable. Note that [54,13,20] use the bounding box annotations during the training, furthermore [54,13] are binary action detection methods. Our method is a weakly-supervised multi-label action recognition/localization approach.

* Action	Subset of frames					All frames				
	[54]	[13]	[20]	[23]	Ours	[54]	[13]	[20]	[23]	Ours
Diving	16.4	36.5	43.4	46.7	42.8	22.6	37.0	N/A	44.3	43.7
Golf	N/A	N/A	37.1	51.3	57.0	N/A	N/A	N/A	50.5	52.3
Kicking	N/A	N/A	36.8	50.6	59.1	N/A	N/A	N/A	48.3	52.9
Lifting	N/A	N/A	68.8	55.0	71.1	N/A	N/A	N/A	51.4	63.5
Hours ride	62.2	68.1	21.9	29.5	41.9	63.1	64.0	N/A	30.6	32.5
Running	50.2	61.4	20.1	34.3	40.5	48.1	61.9	N/A	33.1	30.1
Skating	N/A	N/A	13.0	40.0	45.7	N/A	N/A	N/A	38.5	43.2
Swing bench	N/A	N/A	32.7	54.8	52.1	N/A	N/A	N/A	54.3	57.5
Swing side	N/A	N/A	16.4	19.3	36.6	N/A	N/A	N/A	20.6	44.1
Walking	N/A	N/A	28.3	39.5	42.1	N/A	N/A	N/A	39.0	47.1
Avg.	N/A	N/A	31.8	42.1	49.09	N/A	N/A	N/A	41.0	46.7

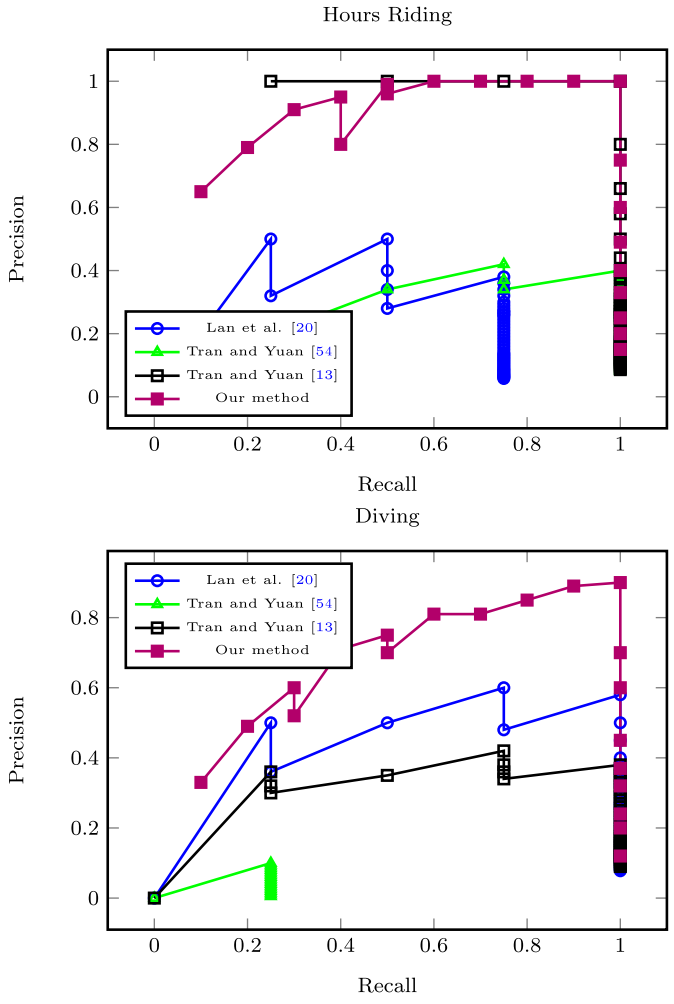


Fig. 13. Precision–recall curves for the two diving and hours riding actions in the UCF Sports dataset. Since [20] only reported results on a subset of frames, in order to be able to compare, we have plotted these curves on the same subset.

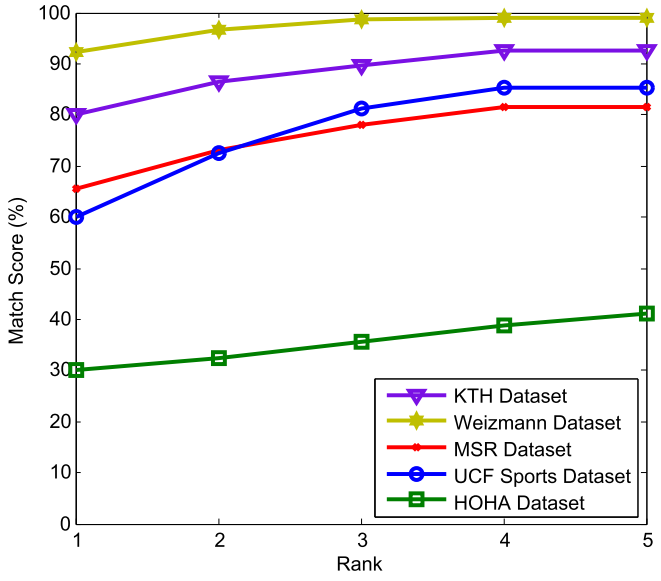


Fig. 14. Cumulative Match Score (CMS) for activity recognition/localization in the KTH, Weizmann, MSR2, HOHA and UCF Sports. The score is calculated for each dataset independently.

each class a representative histogram is extracted. This histogram is used to localize the action of interest in the video, spatio-temporally. Therefore, this approach is working with a weakly supervised multi-label setting. We solve the non-negative matrix completion problem using a convex alternating direction method.

As a direction for the future, it is more desirable to have precise activity locations rather than a bounding box. So, instead of putting the feature vectors for the whole video in the matrix for matrix completion, we might be able to put feature vectors for the parts associated in the video. Furthermore, matrix completion finds linear relations over the instances. One can kernelize the matrix completion framework to get superior results.

References

[1] J. Aggarwal, M. Ryoo, Human activity analysis: a review, ACM Comput. Surv. 43 (3) (2011) 16:1–16:43.
[2] R. Poppe, A survey on vision-based human action recognition, Image Vis. Comput. 28 (6) (2010) 976–990.
[3] O. Maron, T. Lozano-Prez, A framework for multiple-instance learning, Advances in Neural Information Processing Systems, MIT Press 1998, pp. 570–576.
[4] S. Andrews, I. Tsochantaridis, T. Hofmann, Support vector machines for multiple-instance learning, Advances in Neural Information Processing Systems, MIT Press 2003, pp. 561–568.
[5] F. Li, C. Sminchisescu, Convex multiple-instance learning by estimating likelihood ratio, in: J. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R. Zemel, A. Culotta (Eds.), Advances in Neural Information Processing Systems, 23 2010, pp. 1360–1368.
[6] C. Bergeron, G. Moore, J. Zaretzki, C.M. Breneman, K.P. Bennett, Fast bundle algorithm for multiple-instance learning, IEEE Trans. Pattern Anal. Mach. Intell. 34 (6) (2012) 1068–1079.
[7] J. Sivic, A. Zisserman, Video Google: a text retrieval approach to object matching in videos, Proceedings of the Ninth IEEE International Conference on Computer Vision, vol. 2, IEEE Computer Society, Washington, DC, USA 2003, p. 1470.
[8] I. Laptev, M. Marszałek, C. Schmid, B. Rozenfeld, Learning Realistic Human Actions From Movies, In: Conference on Computer Vision & Pattern Recognition, 2008.
[9] H. Wang, M.M. Ullah, A. Kläser, I. Laptev, C. Schmid, Evaluation of local spatio-temporal features for action recognition, British Machine Vision Conference 2009, p. 127.
[10] H. Wang, A. Kläser, C. Schmid, L. Cheng-Lin, Action recognition by dense trajectories, IEEE Conference on Computer Vision & Pattern Recognition, Colorado Springs, United States 2011, pp. 3169–3176.
[11] Y. Tian, R. Sukthankar, M. Shah, Spatiotemporal deformable part models for action detection, IEEE International Conference on Computer Vision and Pattern Recodnition (CVPR), 2013.
[12] A. Kovashka, K. Grauman, Learning a hierarchy of discriminative space–time neighborhood features for human action recognition, IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2010.
[13] D. Tran, J. Yuan, Max-margin structured output regression for spatio-temporal action localization, Advances in Neural Information Processing Systems (NIPS), 2012.
[14] O. Duchenne, I. Laptev, J. Sivic, F. Bach, J. Ponce, Automatic annotation of human action in video, IEEE International Conference on Computer Vision (ICCV), 2009.
[15] D. Tran, J. Yuan, D. Forsyth, Video event detection: From subvolume localization to spatio-temporal path search, IEEE Trans. PAMI.
[16] B.G. Vijay Kumar, I. Patras, Supervised dictionary learning for action localization, 10th IEEE International Conference on Automatic Face and Gesture Recognition (FG) April 2013, pp. 22–26.
[17] A. Gaidon, Z. Harchaoui, C. Schmid, Temporal localization of actions with actoms, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2782–2795.
[18] M. Raptis, I. Kokkinos, S. Soatto, Discovering discriminative action parts from mid-level video representations, IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) 2012, pp. 1242–1249.
[19] M. Hoai, Z. zhong Lan, F. De la Torre, Joint segmentation and classification of human actions in video, IEEE International Conference on Computer Vision and Pattern Recodnition (CVPR), 2011.
[20] T. Lan, Y. Wang, G. Mori, Discriminative figure-centric models for joint action localization and recognition, IEEE International Conference on Computer Vision (ICCV), 2011.
[21] P. Siva, T. Xiang, Weakly supervised action detection, Proceedings of the British Machine Vision Conference, BMVA Press 2011, p. 65.1-65.0.
[22] C.-Y. Chen, K. Grauman, Efficient activity detection with max-subgraph search, IEEE Conference on Computer Vision and Pattern Recognition, 2012.
[23] S. Ma, J. Zhang, N. Ikizler-Cinbis, S. Sclaroff, Action recognition and localization by hierarchical space–time segments, IEEE International Conference on Computer Vision (ICCV), 2013.
[24] M. Hoai, L. Torresani, F.D. la Torre, C. Rother, Learning discriminative localization from weakly labeled data, Pattern Recogn. 47 (3) (2014) 1523–1534.
[25] M.S. Ryoo, J.K. Aggarwal, Spatio-temporal relationship match: video structure comparison for recognition of complex human activities, IEEE International Conference on Computer Vision (ICCV), 2009.
[26] M.D. Rodriguez, J. Ahmed, M. Shah, Action mach: a spatio-temporal maximum average correlation height filter for action recognition, IEEE International Conference on Computer Vision and Pattern Recodnition (CVPR), 2008.

[27] Y. Sheikh, M. Sheikh, M. Shah, Exploring the space of a human action, IEEE International Conference on Computer Vision (ICCV), 2005.
[28] L. Cao, Z. Liu, T.S. Huang, Cross-dataset action detection, IEEE International Conference on Computer Vision and Pattern Recodnition (CVPR) 2010, pp. 1998–2005.
[29] T.G. Dietterich, R.H. Lathrop, T. Lozano-Pérez, Solving the multiple instance problem with axis-parallel rectangles, Artif. Intell. 89 (1–2) (1997) 31–71.
[30] R.S. Cabral, F. De la Torre, J.P. Costeira, A. Bernardino, Matrix completion for multi-label image classification, Advances in Neural Information Processing Systems (NIPS), 2011.
[31] C. Schuldt, I. Laptev, B. Caputo, Recognizing human actions: a local SVM approach, ICPR'04, Volume 3 - Volume 03, IEEE Computer Society, Washington, DC, USA 2004, pp. 32–36.
[32] S. Wu, O. Oreifej, M. Shah, Action recognition in videos acquired by a moving camera using motion decomposition of Lagrangian particle trajectories, IEEE International Conference on Computer Vision (ICCV) 2011, pp. 1419–1426.
[33] Heng Wang, Alexander Kläser, Cordelia Schmid, Cheng-Lin Liu, Dense trajectories and motion boundary descriptors for action recognition, International Journal of Computer Vision, Springer Verlag (Germany) 103 (1) (2013) 60–79.
[34] A.F. Bobick, J.W. Davis, The recognition of human movement using temporal templates, IEEE Trans. Pattern Anal. Mach. Intell. 23 (3) (2001) 257–267.
[35] S. Khokhar, I. Saleemi, M. Shah, Multi-agent event recognition by preservation of spatiotemporal relationships between probabilistic models, Image Vis. Comput. 31 (9) (2013) 603–615.
[36] A. Sanin, C. Sanderson, M. Harandi, B. Lovell, Spatio-temporal covariance descriptors for action and gesture recognition, Workshop on the Applications of Computer Vision (WACV), 2013.
[37] J. Yuan, Z. Liu, Y. Wu, Discriminative video pattern search for efficient action detection, IEEE Trans. Pattern Anal. Mach. Intell. 33 (9) (2011) 1728–1743.
[38] K. Tang, L. Fei-Fei, D. Koller, Learning latent temporal structure for complex event detection, IEEE International Conference on Computer Vision and Pattern Recodnition (CVPR), Providence, RI, USA, 2012.
[39] E.J. Candès, B. Recht, Exact matrix completion via convex optimization, Found. Comput. Math. 9 (6) (2009) 717–772.
[40] M. Fazel, Matrix Rank Minimization with Applications (Ph.D. thesis) Stanford University, 2002.
[41] Z. Zhang, Y. Matsushita, Y. Ma, Camera calibration with lens distortion from low-rank textures, IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, Washington, DC, USA 2011, pp. 2321–2328.
[42] Y. Dai, H. Li, M. He, Element-wise factorization for n-view projective reconstruction, Proceedings of the 11th European conference on Computer Vision: Part IV, ECCV'10, Springer-Verlag, Berlin, Heidelberg 2010, pp. 396–409.
[43] B. Cheng, G. Liu, J. Wang, Z. Huang, S. Yan, Multi-task low-rank affinity pursuit for image segmentation, IEEE International Conference on Computer Vision (ICCV) 2011, pp. 2439–2446.
[44] R. Cabral, F. De la Torre, J. Costeira, A. Bernardino, Matrix completion for weakly-supervised multi-label image classification, IEEE Trans. Pattern Anal. Mach. Intell. 37 (1) (2015) 121–135.
[45] A.B. Goldberg, X. Zhu, B. Recht, J.-M. Xu, R.D. Nowak, Transduction with matrix completion: three birds with one stone, Advances in Neural Information Processing Systems (NIPS) 2010, pp. 757–765.
[46] Z. Lin, M. Chen, L. Wu, Y. Ma, The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices, University of Illinois at Urbana–Champaign Technical Report UILU-ENG-09-2215, 2009.
[47] J. Yang, X. Yuan, Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization, Math. Comput. 82 (281) (2013) 301–329.
[48] R. Andreani, J. Martinez, M. Schuverdt, On the relation between constant positive linear dependence condition and quasinormality constraint qualification, J. Optim. Theory Appl. 125 (2005) 473–483.
[49] J.-F. Cai, E.J. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion, SIAM J. Optim. 20 (4) (2010) 1956–1982.
[50] Y. Xu, W. Yin, Z. Wen, Y. Zhang, An alternating direction algorithm for matrix completion with nonnegative factors, Frontiers of Mathematics in China 7 (2) (2012) 365–384.
[51] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1627–1645.
[52] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, IEEE Trans. Pattern Anal. Mach. Intell. 24 (5) (2002) 603–619.
[53] L. Gorelick, M. Blank, E. Shechtman, M. Irani, R. Basri, Actions as space–time shapes, IEEE Trans. Pattern Anal. Mach. Intell. 29 (12) (2007) 2247–2253.
[54] D. Tran, J. Yuan, Optimal spatio-temporal path discovery for video event detection, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2011, pp. 3321–3328.
[55] S. Savarese, A. DelPozo, J.C. Niebles, L. Fei-Fei, Spatial-temporal correlations for unsupervised action classification, Proceedings of the 2008 IEEE Workshop on Motion and Video Computing, WMVC '08, IEEE Computer Society, Washington, DC, USA 2008, pp. 1–8.
[56] J.C. Niebles, H. Wang, L. Fei-Fei, Unsupervised learning of human action categories using spatial-temporal words, Int. J. Comput. Vis. 79 (3) (2008) 299–318.
[57] A. Fathi, G. Mori, Action recognition by learning mid-level motion features, IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2008.
[58] J. Liu, J. Luo, M. Shah, Recognizing Realistic Actions from Videos "In The Wild", 2009.
[59] Q.V. Le, W.Y. Zou, S.Y. Yeung, A.Y. Ng, Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis, IEEE Conference on Computer Vision and Pattern Recognition 2011, pp. 3361–3368.
[60] Z. Zhang, Y. Hu, S. Chan, L.-T. Chia, Motion context: a new representation for human action recognition, ECCV 2008, vol. 5305 2008, pp. 817–829.
[61] M. Bregonzio, S. Gong, T. Xiang, Recognising action as clouds of space–time interest points, IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) 2009, pp. 1948–1955.
[62] Y. Tian, L. Cao, Z. Liu, Z. Zhang, Hierarchical filtered motion for action recognition in crowded videos, IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 42 (3) (2012) 313–323.
[63] J. Yuan, Z. Liu, Y. Wu, Discriminative subvolume search for efficient action detection, IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2009.
[64] A. Klaeser, M. Marszalek, C. Schmid, A spatio-temporal descriptor based on 3D-gradients, Proceedings of the British Machine Vision Conference, BMVA Press 2008, pp. 99.1–99.10.
[65] P. Matikainen, M. Hebert, R. Sukthankar, Trajectons: action recognition through the motion analysis of tracked features, ICCV Workshops, IEEE 2009, pp. 514–521.
[66] B. Kumar, I. Patras, Learning codebook weights for action detection, Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on 2012, pp. 27–32.
[67] B. Leibe, A. Leonardis, B. Schiele, Combined object categorization and segmentation with an implicit shape model, In ECCV Workshop on Statistical Learning in Computer Vision 2004, pp. 17–32.