Collaborative Sparse Coding for Multiview Action Recognition

Wei Wang and Yan Yan
University of Trento

Luming Zhang and Richang Hong
Hefei University of Technology

Nicu Sebe
University of Trento

Self-similarity matrices (SSMs) show outstanding performance in extracting view-invariant features in multiview action recognition except when there's a large view change. The proposed approach increases SSM robustness by integrating the classifier training process and sparse coding process into a collaborative framework.

Many applications aim to recognize activities; for example, this occurs in human-computer interactive games, search engines, and online video surveillance systems. By automatically annotating actions, videos can be summarized with label information, allowing search engines to make better recommendations (such as when a user searches for "dunks in basketball games"). More recently, a higher-level recognition of human activity was proposed based on *action recognition*—that is, activity involving the interaction between humans and objects in complex scenes.[1] Usually, the same action observed from different viewpoints has considerable differences. Therefore, an efficient method to extract robust view-invariant features is essential for multiview action recognition.

Some feature extraction approaches encode both spatial and temporal information.[2] The video's features can be roughly grouped into two types: 2D or 3D. Many approaches use 3D models to tackle the multiview action recognition problem. First, such models use geometric transitions to obtain projections across different viewpoints. Then, they compare the observations with the projections to find the viewpoint that best matches the observations. Accurately finding body joints to build the 3D model remains an open problem. Additionally, the built model has too many degree-of-freedom parameters, which must be carefully calibrated. Finally, the model requires high-resolution videos to locate body joints and sometimes requires motion-capture data.[3]

An alternative solution for multiview action recognition is to design view-invariant 2D features. Ali Farhadi and Mostafa Tabrizi proposed split-based representations by clustering similar video frames into *splits*—which are clusters of visually similar image frames. The split-based representations can be transferred among different views, because the change dynamics of the multiview videos are the same.[4] Similarly, Imran Junejo and his colleagues employed self-similarity matrices (SSMs) to encode the frame-to-frame relative changes.[5] However, SSMs can view changes robustly only to a certain extent and are sensitive to large viewpoint changes.

To further improve the performance of SSMs, we propose a collaborative sparse coding framework to increase the discriminative capability of the dictionary. We use collaborative filtering (CF)[6] to integrate classifier training with the sparse coding operations. Thus, we obtain a novel collaborative sparse coding framework that optimizes the classifiers and dictionary collaboratively.

The Framework

The underlying intuition of the framework is similar to manifold learning, which is aimed at obtaining a better representation of the data.[7,8] However, the manifold learning approaches suffer from the "out-of-sample" problem. That is, samples that aren't included in the training set can't be embedded into the manifold space. Sparse coding approaches don't have such problems.

The collaborative sparse coding framework provides a tradeoff between the dictionary reconstruction error and the classification error that come from the sparse coding and the logistic classifiers, respectively. The framework makes three key contributions by

1070-986X/16/$33.00 © 2016 IEEE

Published by the IEEE Computer Society

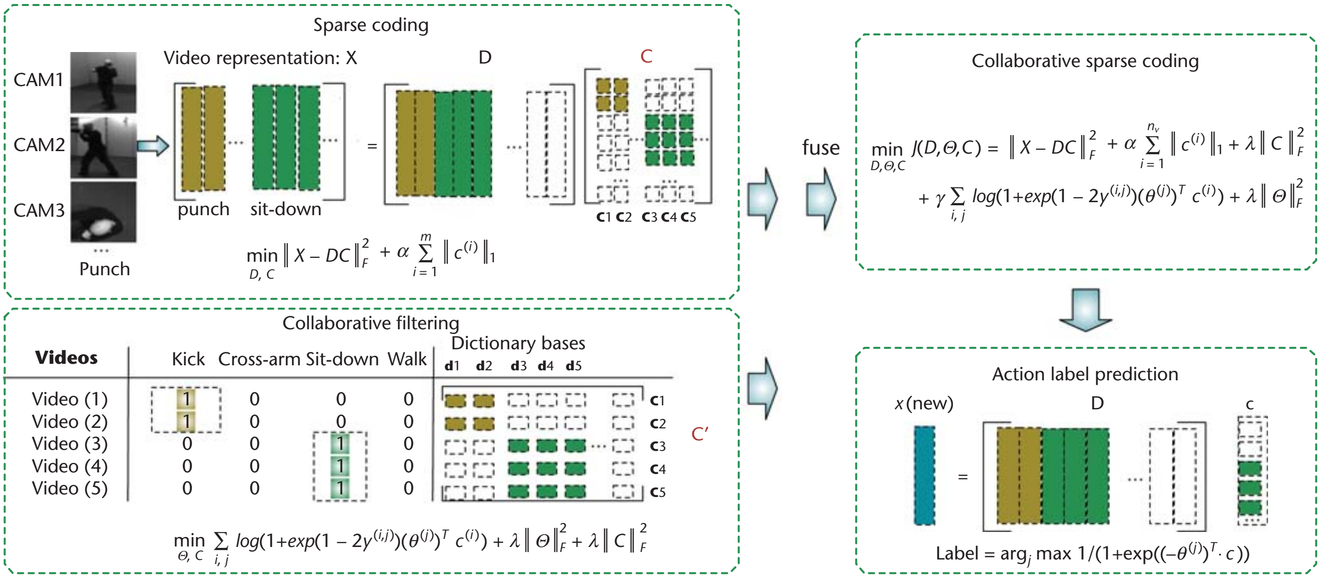

Sparse coding
Video representation: X
D
C
CAM1
CAM2
CAM3
Punch
punch
sit-down
C1 C2 C3 C4 C5
$\min_{D,C}\|X-DC\|_F^2+\alpha\sum_{i=1}^{m}\|c^{(i)}\|_1$
fuse
Collaborative sparse coding
$\min_{D,\Theta,C} J(D,\Theta,C)=\|X-DC\|_F^2+\alpha\sum_{i=1}^{n_v}\|c^{(i)}\|_1+\lambda\|C\|_F^2$
$+\gamma\sum_{i,j} log(1+exp(1-2y^{(i,j)})(\theta^{(j)})^T c^{(i)})+\lambda\|\Theta\|_F^2$
Collaborative filtering
Videos
Kick Cross-arm Sit-down Walk
Dictionary bases
d1 d2 d3 d4 d5
Video (1) 1 0 0 0
Video (2) 1 0 0 0
Video (3) 0 0 1 0
Video (4) 0 0 1 0
Video (5) 0 0 1 0
c1
c2
c3
c4
c5
C′
$\min_{\Theta,C}\sum_{i,j} log(1+exp(1-2y^{(i,j)})(\theta^{(j)})^T c^{(i)})+\lambda\|\Theta\|_F^2+\lambda\|C\|_F^2$
Action label prediction
x (new)
D
c
Label = arg$_j$ max 1/(1+exp((−$\theta^{(j)})^T\cdot c$))

Figure 1. Collaborative sparse coding. The sparse coding and collaborative filtering operations fuse in the collaborative sparse coding framework prior to action label prediction.

- improving the learned dictionary's discriminative capability, creating a more flexible dictionary and robust action classifiers that can be learned jointly;
- applying an efficient algorithm to solve the model; and
- exhibiting an excellent generalization property that can be applied to other pattern recognition tasks.

We now describe our framework, before evaluating its effectiveness using three multiview action recognition datasets.

Sparse Coding

The classical sparse coding model, which aims to minimize the reconstruction error, is defined as

$$\min_{D,C}\|\boldsymbol{X}-\boldsymbol{DC}\|_F^2+\alpha\sum_{i=1}^{m}\|\boldsymbol{c}^{(i)}\|_1,$$

where $\boldsymbol{X}\in\mathbb{R}^{d\times m}$ represents m training data; d is the feature dimension; $\boldsymbol{D}\in\mathbb{R}^{d\times n}$ represents the learned dictionary; n is the number of bases; and $\boldsymbol{C}\in\mathbb{R}^{n\times m}$ is the sparse representation matrix, whose i_{th} column, $c^{(i)}$, is the sparse representation of sample i. The 1_1 norm is a lasso constraint that encourages sparsity, and $\boldsymbol{a}$ balances the reconstruction error and the sparsity penalty.

After learning an over-complete dictionary by minimizing the reconstruction error, the data with the same label can obtain sparse and generative representations. The label information isn't considered in the process. Thus, sparse coding has limited discriminative capability.

Collaborative Filtering

As Figure 1 shows, $\boldsymbol{Y}\in\mathbb{R}^{n_v\times n_a}$ is the video-action rating matrix, where n_v is the number of videos and n_a is the number of actions. The item $y^{(i,j)}$ is either 1 or 0, which denotes whether or not video i belongs to action class j.

The classical CF learns the feature representation $\boldsymbol{c}^{(i)}\in\mathbb{R}^n$, $(\boldsymbol{i}=1,2,\ldots,n_v)$ and the feature preference weight $\theta^{(j)}\in\mathbb{R}^n$, $(j=1,2,\ldots,n_a)$ jointly by minimizing the Euclidean distance between the predicted rating $(\theta^{(j)})^T\boldsymbol{c}^i$and the ground truth $y^{(i,j)}$. The objective function can be put in the following form:

$$\min_{\theta^{(i)},\boldsymbol{c}^{(i)}}\|(\theta^{(j)})^T\boldsymbol{c}^{(i)}-y^{(i,j)}\|^2,$$

where $\theta^{(j)}$ is the column vector in $\Theta_{n\times n_a}$. Action recognition is a multiclass classification problem. Thus, instead of using the classical linear models, we adopt logistic regression models. The output of the logistic function ranges from 0 to 1, and it represents the probability that a video belongs to an action class. The objective function for CF is as follows:

$$\min_{\Theta,\boldsymbol{C}}\ \log(1+\exp(1-2y^{(i,j)})(\theta^{(j)})^T\boldsymbol{c}^{(i)})$$
$$+\lambda\|\Theta\|_F^2+\lambda\|\boldsymbol{C}\|_F^2, \tag{1}$$

where $\|\Theta\|_F^2$ and $\|\boldsymbol{C}\|_F^2$ are the Frobenius norms of Θ and $\boldsymbol{C}$, respectively. Equation 1 is very

```
Initialization: $D \leftarrow D_0$, $C \leftarrow C_0$, $\Theta \leftarrow \Theta_0$,
  repeat
      fix D,Θ,update C
          ratio ← 1
          while ratio>threshold
              run FISTA (modified)
              update ratio
          end while
      fix D,C,update Θ
          parallel gradient descent
      fix C,Θ,update D
          least-squares solution
   until converges
```

Figure 2. Algorithm 1: Solution Structure.

similar to the objective function of multitask learning. Actually, when $\boldsymbol{C}$ is fixed, Equation 1 is a multitask learning model with respect to Θ. When Θ is fixed, it is a multitask learning model with respect to $\boldsymbol{C}$.

Collaborative Sparse Coding

The dictionary and classifiers are usually trained separately. Here, we integrate them together into our model. The learned sparse representation is expected to be view-invariant and discriminative at the same time. The optimal $\boldsymbol{D}$, Θ, $\boldsymbol{C}$ can be obtained by solving the following collaborative sparse coding model:

$$\begin{aligned}\min_{\boldsymbol{D},\Theta,\boldsymbol{C}} \mathrm{J}(\boldsymbol{D},\Theta,\boldsymbol{C}) &= \|\boldsymbol{X}-\boldsymbol{DC}\|_{\mathrm{F}}^2 \\ &+\alpha\sum_{\mathrm{i}=1}^{\mathrm{n_v}}\|\boldsymbol{c}^{(\mathrm{i})}\|_1+\lambda\|\boldsymbol{C}\|_{\mathrm{F}}^2 \\ &+\gamma\sum_{\mathrm{i,j}}\log(1+\exp(1-2\mathrm{y}^{(\mathrm{i,j})})(\theta^{(\mathrm{j})})^{\mathrm{T}}\boldsymbol{c}^{(\mathrm{i})}) \\ &+\lambda\|\Theta\|_{\mathrm{F}}^2,\end{aligned} \tag{2}$$

where γ balances the sparse coding loss and the classification loss. By compromising the sparse coding loss and classification loss, a robust dictionary $\boldsymbol{D}$ and a group of action classifiers Θ can be learned jointly.

Label prediction. As Figure 1 shows, the label prediction for a new test video $\boldsymbol{x}$ consists of two steps. First, the sparse representation $\boldsymbol{c}$ of the new video is calculated by solving the following optimization problem:

$$\min_{\boldsymbol{c}}\|\boldsymbol{x}-\boldsymbol{Dc}\|_{\mathrm{F}}^2+\alpha\|\boldsymbol{c}\|_1.$$

Then, the probability that the new video belongs to action class j is

$$\mathrm{p}(\mathrm{y}=1|\boldsymbol{c},\theta^{(\mathrm{j})})=\mathrm{h}_{\theta^{(\mathrm{j})}}(\boldsymbol{c}).$$

The predicted label is the one that maximizes $\mathrm{h}_{\theta^{(\mathrm{j})}}(\boldsymbol{c})$:

$$\mathrm{label}=\arg_{\mathrm{j}}\max 1/(1+\exp((-\theta^{(\mathrm{j})})^{\mathrm{T}}\cdot\boldsymbol{c})).$$

Optimization. The collaborative sparse coding model is an optimization problem. We propose the structure in Algorithm 1 to solve this problem (see Figure 2). The gradient descent method is employed to solve the model in Equation 2. When only one variable is left to optimize and the rest are fixed, the problem becomes convex. Thus, we optimize the variables alternatively by fixing the rest.

Initialization in Algorithm 1 occurs as follows. We employ *k*-means clustering to find *k* centroids as the initial bases in dictionary $\boldsymbol{D}_0$ and $\boldsymbol{C}_0$ are set to $\boldsymbol{0}$. The *loop* in Algorithm 1 consists of three parts.

The first loop is *Fix* $\boldsymbol{C}$, Θ, *update* $\boldsymbol{D}$. In Equation 2, only the first term is related to $\boldsymbol{D}$, and it is a least square problem. By setting the derivative of Equation 2 equal to $\boldsymbol{0}$ with respect to $\boldsymbol{D}$, the following equation is obtained:

$$(\boldsymbol{DC}-\mathbf{X})\boldsymbol{C}^{\mathrm{T}}=0\Rightarrow\boldsymbol{D}=\mathbf{X}\boldsymbol{C}^{\mathrm{T}}(\boldsymbol{CC}^{\mathrm{T}})^{-1}.$$

We employ the following equation to update $\boldsymbol{D}$:

$$\boldsymbol{D}=\boldsymbol{XC}^{\mathrm{T}}(\boldsymbol{CC}^{\mathrm{T}}+\lambda\boldsymbol{I})^{-1},$$

where $\boldsymbol{\lambda}$ is a small constant and guarantees that the matrix $\boldsymbol{CC}^{\mathrm{T}}+\lambda\boldsymbol{I}$ is invertible in case $\boldsymbol{CC}^{\mathrm{T}}$ is singular.

The second loop is *Fix* $\boldsymbol{C}$, $\boldsymbol{D}$, *update* Θ. When $\boldsymbol{C}$ and $\boldsymbol{D}$ are fixed, we employ the parallel gradient descent method to tackle the problem, which is formulated as

$$\theta^{(\mathrm{j})}=\theta^{(\mathrm{j})}-\delta\frac{\partial}{\partial\theta^{(\mathrm{j})}}\mathrm{J}.$$

The derivative with respect to $\theta^{(\mathrm{j})}$ is

$$\frac{\partial}{\partial\theta^{(\mathrm{j})}}\mathrm{J}=\sum_{\mathrm{i}=1}^{\mathrm{n_v}}(\mathrm{h}_{\theta^{(\mathrm{j})}}(c^{(\mathrm{i})})-\mathrm{y}^{(\mathrm{i,j})})c^{(\mathrm{i})}+2\lambda\theta^{(\mathrm{j})}.$$

Because $\theta^{(\mathrm{j})}$ are independent from each other, we optimize them in parallel.

The third loop is *Fix* Θ, $\boldsymbol{D}$, *update* $\boldsymbol{C}$. Because **Θ** and $\boldsymbol{D}$ are fixed, the fifth term is a constant. By removing the constant term, the objective function in Equation 2 is equivalent to

$$\begin{aligned}\min_{\boldsymbol{C}}\mathrm{L}(\boldsymbol{C};\boldsymbol{X})&=\|\boldsymbol{DC}-\boldsymbol{X}\|_{\mathrm{F}}^2+\alpha\|\boldsymbol{C}\|_1\\&+\lambda\sum_{\mathrm{s}=1}^{\mathrm{K}}\|\boldsymbol{C}\|_{\mathrm{F}}^2+\gamma\sum_{\mathrm{i,j}}\log(1\\&+\exp(1-2\mathrm{y}^{(\mathrm{i,j})})(\theta^{(\mathrm{j})})^{\mathrm{T}}\boldsymbol{c}^{(\mathrm{i})}).\end{aligned}$$

Amir Beck and Marc Teboulle proposed the Fast Iterative Soft-Thresholding Algorithm

(FISTA) to solve the classical sparse coding problem.[9] A soft-threshold step is incorporated into FISTA to guarantee the sparseness of the solution; the FISTA algorithm then can be modified to tackle the problem. In the classical dictionary learning model, the sparse representations of training data are independent from each other. Thus, each $c^{(i)}$ can be optimized independently in parallel. The sub-objective in our model is

$$\begin{aligned}&\|\boldsymbol{D}\boldsymbol{c}^{(\mathrm{i})}-\boldsymbol{x}^{(\mathrm{i})}\|^2\\&\quad+\gamma\sum_{\mathrm{j}}\log(1+\exp(1-2\mathrm{y}^{(\mathrm{i,j})})(\theta^{(\mathrm{j})})^{\mathrm{T}}\boldsymbol{c}^{(\mathrm{i})})\\&\quad+\alpha\|\boldsymbol{c}^{(\mathrm{i})}\|_1+\lambda\|\boldsymbol{c}^{(\mathrm{i})}\|_{\mathrm{F}}^2.\end{aligned}$$

For training data $\boldsymbol{x}^{(\mathrm{i})}\in\boldsymbol{X}$ in the equation above, its sparse representation $\boldsymbol{c}^{(\mathrm{i})}$ ($\boldsymbol{c}^{(\mathrm{i})}\in\boldsymbol{C}$) is also independent from other $\boldsymbol{c}^{(\mathrm{k})}$ ($\boldsymbol{c}^{(\mathrm{k})}\in\boldsymbol{C}$). Thus, we modify the FISTA algorithm to optimize the new objectives in parallel. The gradient of $\boldsymbol{c}^{(\mathrm{i})}$ is

$$\begin{aligned}\frac{\partial L}{\partial\boldsymbol{c}^{(\mathrm{i})}}=&2\boldsymbol{D}^{\mathrm{T}}(\boldsymbol{D}\boldsymbol{c}^{(\mathrm{i})}-\boldsymbol{x}^{(\mathrm{i})})\\&+\gamma[1/\Big(1+\exp(-(\theta^{(\mathrm{j})})^{\mathrm{T}}\boldsymbol{c}^{(\mathrm{i})})\Big)-\mathrm{y}^{(\mathrm{i,j})}]\theta^{(\mathrm{j})}\\&+2\lambda\boldsymbol{c}^{(\mathrm{i})}.\end{aligned}$$

Then, when updating $\boldsymbol{C}$, all $\boldsymbol{c}^{(\mathrm{i})}\in\boldsymbol{C}$ are updated in parallel using the following form:

$$\boldsymbol{c}^{(\mathrm{i})}:=\boldsymbol{c}^{(\mathrm{i})}-\delta\frac{\partial \mathrm{L}}{\partial\boldsymbol{c}^{(\mathrm{i})}}.$$

This updating procedure of $\boldsymbol{C}$ will repeat until it converges.

Experiments

In the experiments, we use three widely used multiview action recognition datasets to evaluate our framework. A comparison with other baselines demonstrates the superior performance of our framework.

Dataset

As Figure 3 shows, we evaluated our approach on three public multiview action recognition datasets: the INRIA Xmas Motion Acquisition Sequences (IXMAS) dataset,[10] the newer IXMAS (NIXMAS) dataset, and the Occluded INRIA Xmas Motion Acquisition Sequences (OIXMAS) dataset.[11] The IXMAS dataset consists of 12 action classes: *check watch, cross arms, scratch head, sit down, get up, turn around, walk, wave, punch, kick, point,* and *pick up*. Each action is performed three times by 11 actors and is recorded by five cameras that observe the actions from five different viewpoints. The NIXMAS dataset is recorded with different actors, cameras, and viewpoints; about two-thirds of the videos have objects that partially occlude the actors. Overall, NIXMAS contains 1,148 sequences. The OIXMAS dataset contains the same actions as IXMAS, but the actions are performed by different actors, who could be partially occluded.

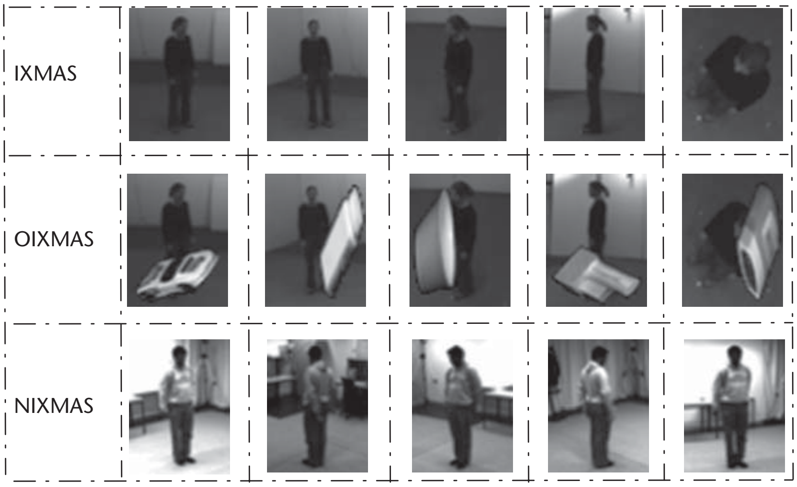


Figure 3. Examples from the three multiview action recognition datasets. All the actions are recorded by five cameras from different viewpoints, and the actors in OIXMAS are partially occluded.

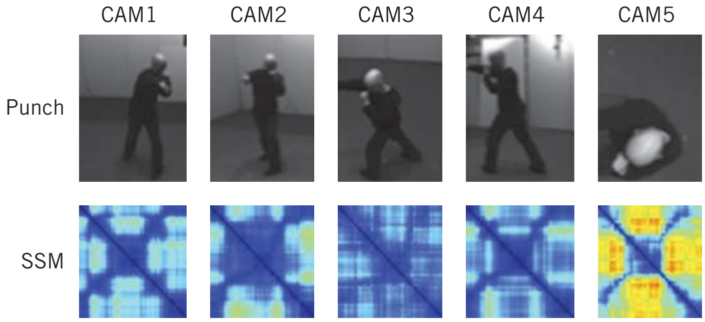
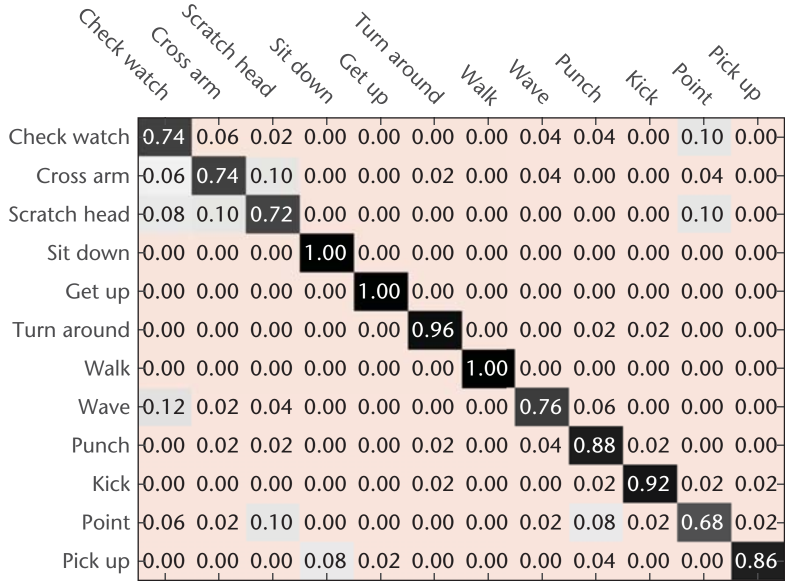


Figure 4. Examples from the IXMAS dataset and the extracted SSM feature. The SSM features for the actions from the four side cameras are visually similar, while the SSM feature from the ceiling camera is quite different.

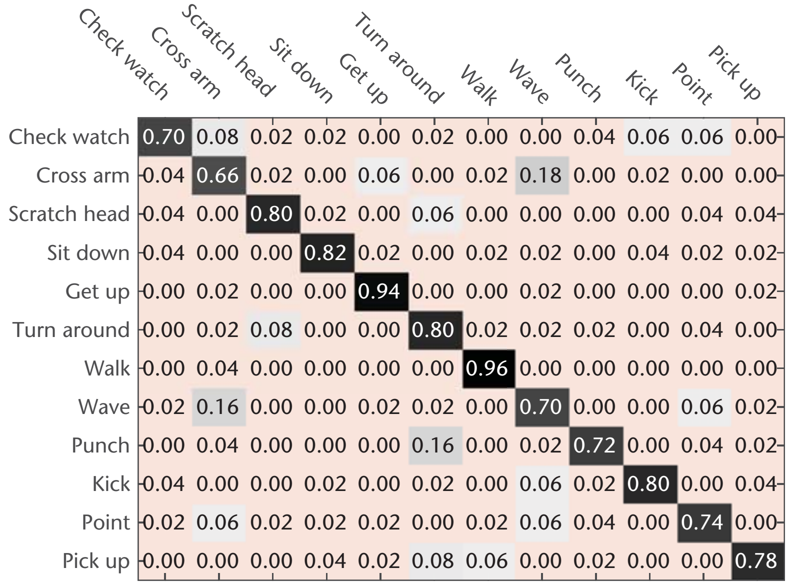
Implementation Details

As discussed before, the collaborative sparse coding is based on SSM descriptors using histogram of oriented gradients (HOG) and histogram of optical flow (HOF) features to describe each individual frame. Figure 4 shows an example from the IXMAS dataset and the corresponding extracted SSM feature. From Figure 4, we can observe that the extracted SSM from



	Check watch	Cross arm	Scratch head	Sit down	Get up	Turn around	Walk	Wave	Punch	Kick	Point	Pick up
Check watch	0.74	0.06	0.02	0.00	0.00	0.00	0.00	0.04	0.04	0.00	0.10	0.00
Cross arm	0.06	0.74	0.10	0.00	0.00	0.02	0.00	0.04	0.00	0.00	0.04	0.00
Scratch head	0.08	0.10	0.72	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00
Sit down	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Get up	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Turn around	0.00	0.00	0.00	0.00	0.00	0.96	0.00	0.00	0.02	0.02	0.00	0.00
Walk	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
Wave	0.12	0.02	0.04	0.00	0.00	0.00	0.00	0.76	0.06	0.00	0.00	0.00
Punch	0.00	0.02	0.02	0.00	0.00	0.02	0.00	0.04	0.88	0.02	0.00	0.00
Kick	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.02	0.92	0.02	0.02
Point	0.06	0.02	0.10	0.00	0.00	0.00	0.00	0.02	0.08	0.02	0.68	0.02
Pick up	0.00	0.00	0.00	0.08	0.02	0.00	0.00	0.00	0.04	0.00	0.00	0.86

***Figure 5.** The confusion matrix of the collaborative sparse coding approach for the IXMAS dataset. The darker shading represents a larger value. The boxes along the diagonal represents the precisions of the actions (row labels), while the rest of the boxes represent the false positive predictions for the actions (column labels). We can observe that very higher precisions can be achieved for most of the actions, except for "point," which is sometimes labelled as "check-watch" and "punch."*



	Check watch	Cross arm	Scratch head	Sit down	Get up	Turn around	Walk	Wave	Punch	Kick	Point	Pick up
Check watch	0.70	0.08	0.02	0.02	0.00	0.02	0.00	0.00	0.04	0.06	0.06	0.00
Cross arm	0.04	0.66	0.02	0.00	0.06	0.00	0.02	0.18	0.00	0.02	0.00	0.00
Scratch head	0.04	0.00	0.80	0.02	0.00	0.06	0.00	0.00	0.00	0.00	0.04	0.04
Sit down	0.04	0.00	0.00	0.82	0.02	0.00	0.02	0.02	0.00	0.04	0.02	0.02
Get up	0.00	0.02	0.00	0.00	0.94	0.00	0.00	0.02	0.00	0.00	0.00	0.02
Turn around	0.00	0.02	0.08	0.00	0.00	0.80	0.02	0.02	0.02	0.00	0.04	0.00
Walk	0.00	0.04	0.00	0.00	0.00	0.00	0.96	0.00	0.00	0.00	0.00	0.00
Wave	0.02	0.16	0.00	0.00	0.02	0.02	0.00	0.70	0.00	0.00	0.06	0.02
Punch	0.00	0.04	0.00	0.00	0.00	0.16	0.00	0.02	0.72	0.00	0.04	0.02
Kick	0.04	0.00	0.00	0.02	0.00	0.02	0.00	0.06	0.02	0.80	0.00	0.04
Point	0.02	0.06	0.02	0.02	0.02	0.00	0.02	0.06	0.04	0.00	0.74	0.00
Pick up	0.00	0.00	0.00	0.04	0.02	0.08	0.06	0.00	0.02	0.00	0.00	0.78

***Figure 6.** The confusion matrix of the collaborative sparse coding approach for the OIXMAS dataset. This dataset is more challenging as the actors are partially occluded. Thus, on average, the performance on this dataset is inferior compared with the un-occluded IXMAS dataset.*

camera 5 (CAM5, on the ceiling) is quite different from the SSMs extracted from the four side cameras. In our experiments, all regularization parameters $\boldsymbol{\alpha}$, γ, and λ are tuned from $[10^{-3}, 10^{-2}, \ldots, 10^{3}]$.

We employ two settings for the experiment: the multiview setting and the cross-view setting. For the *multiview setting,* we have access to the videos from all viewpoints for training. For the *cross-view setting,* one camera view is missing in the training data, and we use the model learned from the other four camera views to perform prediction for the missing view.

Experimental Results

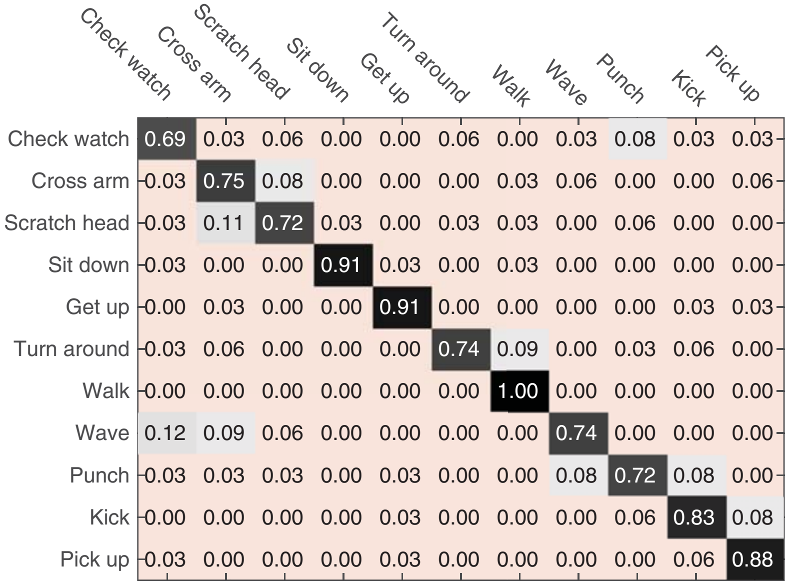
We compare the proposed approach with a radial basis kernel Support Vector Machine (SVM)[5] and the multitask learning approach (MTL),[12] which assumes that all tasks are related to each other. We also combine the sparse coding with SVM (SC + SVM) and MTL (SC + MTL) as baselines.

Multiview action recognition. For the multiview setting, we use the standard two-thirds and one-third split for training and testing. Figures 5–7 show the confusion matrices of our proposed collaborative sparse coding approach for the IXMAS, OIXMAS, and NIXMAS datasets, respectively. In Figure 5, it is interesting to observe that for actions such as `get up`, `walk`, and `turn-around`, our approach achieves very high recognition accuracies. However, some actions (such as `point`) have relatively poor performances. The reason for this is that the action `point` is visually similar to three other actions: `scratch head`, `punch`, and `check watch`. All four actions involve the movement of hands. In contrast, `sit down`, `get up`, and `walk` are visually distinct. Thus, it is more likely for the framework to learn discriminative representations from these actions. Therefore, relatively better performances are achieved on these actions.

Table 1 shows the mean action recognition accuracy of all the cameras for different approaches.[13–16] As the table shows, the baselines SC+SVM and SC+MTL outperform SVM and MTL alone, respectively. This indicates that if we use sparse coding to do preprocessing, the performance of the classifiers can be improved. However, the improvement is very limited, because the dictionary and classifiers are trained separately. Our approach optimizes the classifiers and dictionary iteratively, which leads to a dramatic performance improvement. Figure 8 shows some qualitative results on the IXMAS dataset for our proposed collaborative dictionary learning approach and multitask learning approach for multiview action recognition.

We also analyzed the sensitivity of each parameter using the IXMAS dataset. The optimal mean action recognition accuracy can be obtained when $\gamma = 10$, $\boldsymbol{a} = 0.1$, and $\lambda = 1$. By fixing $\boldsymbol{a}$ and $\boldsymbol{\lambda}$ to the optimal values, the change of γ can lead to a huge performance difference; the largest performance difference is 0.06. Next, we tested the sensitivity of $\boldsymbol{a}$ and $\boldsymbol{\lambda}$ by fixing the rest parameters to the optimal value. The largest performance differences for $\boldsymbol{a}$ and λ are 0.02 and 0.0015. Thus, the sensitivity rank of the parameters is as follows: $\gamma > \boldsymbol{a} > \lambda$.

Cross-view action recognition. Tables 2–4 show the performances of different approaches on the IXMAS, OIXMAS, and NIXMAS datasets. Our approach achieves better average performance compared with other baselines, which shows the effectiveness of our learned class-wise dictionary. It is also interesting to note that the fifth camera always has low action recognition accuracy regardless of the classification method. One reasonable explanation is that the fifth camera is placed on the ceiling, and the motion dynamics of different actions appear visually similar to each other from this viewpoint.



	Check watch	Cross arm	Scratch head	Sit down	Get up	Turn around	Walk	Wave	Punch	Kick	Pick up
Check watch	0.69	0.03	0.06	0.00	0.00	0.06	0.00	0.03	0.08	0.03	0.03
Cross arm	0.03	0.75	0.08	0.00	0.00	0.00	0.03	0.06	0.00	0.00	0.06
Scratch head	0.03	0.11	0.72	0.03	0.00	0.03	0.03	0.00	0.06	0.00	0.00
Sit down	0.03	0.00	0.00	0.91	0.03	0.00	0.03	0.00	0.00	0.00	0.00
Get up	0.00	0.03	0.00	0.00	0.91	0.00	0.00	0.00	0.00	0.03	0.03
Turn around	0.03	0.06	0.00	0.00	0.00	0.74	0.09	0.00	0.03	0.06	0.00
Walk	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
Wave	0.12	0.09	0.06	0.00	0.00	0.00	0.00	0.74	0.00	0.00	0.00
Punch	0.03	0.03	0.03	0.00	0.03	0.00	0.00	0.08	0.72	0.08	0.00
Kick	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.06	0.83	0.08
Pick up	0.03	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.06	0.88

Figure 7. The confusion matrix of the collaborative sparse coding approach for the NIXMAS dataset. The NIXMAS contains less occluded actions than the OIXMAS dataset. Thus, better performance can be achieved on this dataset. But its performance is still inferior when compared with the un-occluded IXMAS dataset.

Extensive experimental results illustrate that our proposed method outperforms other important baselines for multiview action recognition. In the future work, we will consider the correlation between the classifiers, as well as the manifold information of the data. For example, we can suppress the urge of feature sharing between classifiers by adding a 1_1 norm penalty to the classifier parameters.

Our proposed collaborative sparse coding model can also be applied for other tasks, such

Table 1. Multiview action recognition accuracy of different approaches for three datasets.

	IXMAS*	OIXMAS†	NIXMAS‡
Support Vector Machine (SVM)[5]	0.6425	0.4809	0.5680
Space coding and SVM (SC+SVM)	0.6537	0.5235	0.6206
Multitask learning approach (MTL)[12]	0.6883	0.5608	0.6163
SC+MTL	0.6889	0.6082	0.6228
Ruonan Li and colleagues[13]	0.8120	—	—
Yan Yan and colleagues[12] (earlier work involving some of us—Yan and Sebe)	0.8430	—	—
Ali Farhadi and colleagues[4]	0.5810	—	—
Chunhao Huang and colleagues[14]	0.5730	—	—
Jingen Liu and colleagues[15]	0.7380	—	—
Kishore Reddy and colleagues[16]	0.7260	—	—
Proposed approach	0.8496	0.7516	0.7897

*INRIA Xmas Motion Acquisition Sequences
†Occluded INRIA Xmas Motion Acquisition Sequences
‡Newer IXMAS

Table 2. Cross-view action recognition performance of different approaches on the IXMAS dataset.

	Missing viewpoints					
Methods	**Cam1**	**Cam2**	**Cam3**	**Cam4**	**Cam5**	**Average**
Support Vector Machine (SVM)[5]	0.6663	0.6554	0.6500	0.6243	0.4963	0.6185
Space coding and SVM (SC+SVM)	0.6880	0.6577	0.6701	0.6187	0.5110	0.6291
Multitask learning approach (MTL)[12]	0.7554	0.7462	0.7710	0.6973	0.6332	0.7206
SC+MTL	0.7559	0.8257	0.8003	0.7759	0.6417	0.7599
Proposed method	0.8187	0.8206	0.8048	0.7848	0.7299	0.7918

Table 3. Cross-view action recognition performance of different approaches on the OIXMAS dataset.

	Missing viewpoints					
Methods	**Cam1**	**Cam2**	**Cam3**	**Cam4**	**Cam5**	**Average**
Support Vector Machine (SVM)[5]	0.5639	0.6250	0.5472	0.4677	0.4423	0.5259
Space coding and SVM (SC+SVM)	0.5688	0.6477	0.6001	0.5087	0.4511	0.5553
Multitask learning approach (MTL)[12]	0.5422	0.6540	0.5070	0.5171	0.4730	0.5387
SC+MTL	0.5535	0.6826	0.5366	0.5401	0.4867	0.5599
Proposed method	0.5972	0.6974	0.6434	0.6853	0.5838	0.6414

Table 4. Cross-view action recognition performance of different approaches on the NIXMAS dataset.

	Missing viewpoints					
Methods	**Cam1**	**Cam2**	**Cam3**	**Cam4**	**Cam5**	**Average**
Support Vector Machine (SVM)[5]	0.6410	0.6532	0.5912	0.5924	0.5332	0.6020
Space coding and SVM (SC+SVM)	0.6759	0.6951	0.6226	0.6387	0.5560	0.6377
Multitask learning approach (MTL)[12]	0.7170	0.6993	0.7542	0.6911	0.6792	0.7082
SC+MTL	0.7198	0.7391	0.7559	0.7176	0.6879	0.7240
Proposed method	0.7902	0.7941	0.7527	0.7334	0.6988	0.7533

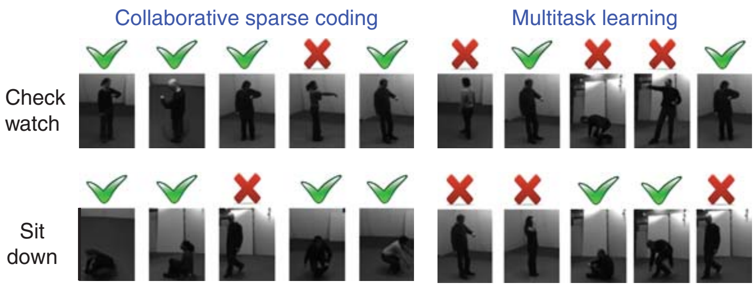


Figure 8. Qualitative results of the proposed approach on the IXMAS dataset. The symbol “√” represents a correct label prediction, while “X” means the wrong label prediction. We can observe that the introduced collaborative sparse coding framework can make more accurate predictions than the multitask learning framework.

as attribute detection.[17,18] Usually, a single attribute can belong to several different categories. For example, the categories *cat* and *dog* all contain the attribute *furry.* Thus, we can regard the categories as the views and treat the attributes as actions. Then the model can be delivered directly to the attribute detection task. **MM**

References

1. J.F. Hu et al., “Recognising Human-Object Interaction via Exemplar Based Modelling,” *Proc. IEEE Int’l Conf. Computer Vision*, 2013, pp. 3144–3151.
2. D. Weinland, R. Ronfard, and E. Boyer, “A Survey of Vision-Based Methods for Action Representation, Segmentation and Recognition,” *Computer Vision*

and Image Understanding, vol. 115, no. 2, 2011, pp. 224–241.

3. P. Peursum, S. Venkatesh, and G. West, “Tracking-as-Recognition for Articulated Full-Body Human Motion Analysis,” *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR), 2007, pp. 1–8.
4. A. Farhadi and M.K. Tabrizi, “Learning to Recognize Activities from the Wrong View Point,” *Computer Vision–ECCV*, LNCS 5302, Springer-Verlag, 2008, pp. 154–166.
5. I.N. Junejo et al., “View-Independent Action Recognition from Temporal Self-Similarities,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, 2011, pp. 172–185.
6. L. Terveen and W. Hill, “Beyond Recommender Systems: Helping People Help Each Other,” *HCI in the New Millennium*, vol. 1, 2001, pp. 487–509.
7. Z. Cui et al., “Generalized Unsupervised Manifold Alignment,” *Advances in Neural Information Processing Systems*, vol. 27, 2014, pp. 2429–2437.
8. Y. Wei et al., “Modality-Dependent Cross-Media Retrieval,” *ACM Trans. Intelligent Systems and Technology*, vol. 7, no. 4, 2015, article no. 57.
9. A. Beck and M. Teboulle, “A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems,” *SIAM J. Imaging Sciences*, vol. 2, no. 1, 2009, pp. 183–202.
10. D. Weinland, R. Ronfard, and E. Boyer, “Free Viewpoint Action Recognition Using Motion History Volumes,” *Computer Vision and Image Understanding*, vol. 104, no. 2, 2006, pp. 249–257.
11. D. Weinland, M. Özuysal, and P. Fua, “Making Action Recognition Robust to Occlusions and Viewpoint Changes,” *Computer Vision–ECCV*, LNCS 6313, Springer-Verlag, 2010, pp. 635–648.
12. Y. Yan et al., “Multitask Linear Discriminant Analysis for View Invariant Action Recognition,” *IEEE Trans. Image Processing*, vol. 23, no. 12, 2014, pp. 5599–5611.
13. R. Li and Todd Zickler, “Discriminative Virtual Views for Cross-View Action Recognition,” *IEEE Conf. Computer Vision and Pattern Recognition* (CVPR), 2012, pp. 2855–2862.
14. C.H. Huang et al., “Recognizing Actions across Cameras by Exploring the Correlated Subspace,” *Computer Vision–ECCV*, LNCS 7583, Springer-Verlag, 2012, pp. 342–351.
15. J. Liu and M. Shah, “Learning Human Actions via Information Maximization,” *IEEE Conf. Computer Vision and Pattern Recognition* (CVPR), 2008, pp. 1–8.
16. K.K. Reddy, J. Liu, and M. Shah, “Incremental Action Recognition Using Feature-Tree,” *IEEE 12th Int’l Conf. Computer Vision*, 2009, pp. 1010–1017.
17. W. Wang et al., “Category Specific Dictionary Learning for Attribute Specific Feature Selection,” *IEEE Trans. Image Processing*, vol. 3, no. 25, 2016, pp. 1465–1478.
18. W. Wang, Y. Yan, and N. Sebe, “Attribute Guided Dictionary Learning,” *Proc 5th ACM Int’l Conf. Multimedia Retrieval*, 2015, pp. 211–218.

Wei Wang is a PhD student in the Multimedia and Human Understanding Group at the University of Trento, Italy. His research interests include machine learning and its application to computer vision and multimedia analysis. Wang received an MS in mechatronics from the University of Southern Denmark. Contact him at wei.wang@unitn.it.

Yan Yan is a research fellow with the Multimedia and Human Understanding Group (MHUG) group at the University of Trento, Italy. His research interests include computer vision, machine learning, and multimedia. Yan received a PhD in computer science from the University of Trento, Italy. He received the Best Student Paper Award at the 2014 International Conference on Pattern Recognition and best paper candidate at ACM Multimedia 2015. Contact him at yan@disi.unitn.it.

Luming Zhang is a professor in the Department of Computer and Information at the Hefei University of Technology, China. His research interests include multimedia analysis, image enhancement, and pattern recognition. Zhang received a PhD in computer science from Zhejiang University, China. Contact him at zglumg@gmail.com.

Richang Hong is a professor in the Department of Computer and Information at the Hefei University of Technology, China. His research interests include multimedia question answering, video content analysis, and pattern recognition. Hong received a PhD in computer science from the University of Science and Technology of China, Hefei. He is a member of the ACM. Contact him at hongrc.hfut@gmail.com.

Nicu Sebe is a professor in the Department Computer Science at the University of Trento, where he leads the research on multimedia information retrieval and human-computer interaction in computer vision applications. Sebe received a PhD in computer science from Leiden University, The Netherlands. He is a senior member of IEEE and the ACM and a fellow of the International Association for Pattern Recognition. Contact him at sebe@disi.unitn.it.