Deeply Learned View-Invariant Features for Cross-View Action Recognition

Yu Kong, *Member, IEEE*, Zhengming Ding, *Student Member, IEEE*,
Jun Li, *Member, IEEE*, and Yun Fu, *Senior Member, IEEE*

***Abstract*— Classifying human actions from varied views is challenging due to huge data variations in different views. The key to this problem is to learn discriminative view-invariant features robust to view variations. In this paper, we address this problem by learning view-specific and view-shared features using novel deep models. View-specific features capture unique dynamics of each view while view-shared features encode common patterns across views. A novel sample-affinity matrix is introduced in learning shared features, which accurately balances information transfer within the samples from multiple views and limits the transfer across samples. This allows us to learn more discriminative shared features robust to view variations. In addition, the incoherence between the two types of features is encouraged to reduce information redundancy and exploit discriminative information in them separately. The discriminative power of the learned features is further improved by encouraging features in the same categories to be geometrically closer. Robust view-invariant features are finally learned by stacking several layers of features. Experimental results on three multi-view data sets show that our approaches outperform the state-of-the-art approaches.**

***Index Terms*— Action recognition, autoencoder, multi-view learning, view-invariant features.**

I. Introduction

HUMAN action data are ubiquitous and are of interest to machine learning [1], [2] and computer vision communities [3], [4]. Generally, action data can be observed from multiple views, for example, dynamic human actions captured by multiple sensor views and various camera views, etc, (Figure 1). Classification on such action data in *cross-view* scenario is challenging as the raw data are captured by various sensor devices at different physical locations, and may appear completely different. For example, in Figure 1(b), an action observed from side view is visually different from the one observed from top view. Therefore, using the features extracted in one view is less discriminative for classifying actions in another view.

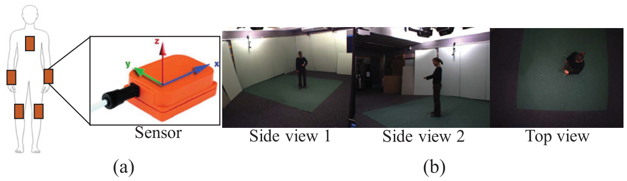


Fig. 1. Examples of multi-view scenarios. (a) Multi-sensor-view, where multiple sensors (orange rectangles) are attached to torso, arms and legs, and human action data are recorded by these sensors. (b) Multi-camera-view, where human actions are recorded by multiple cameras at various viewpoints.

A line of work has been studied to build view-invariant representations for action recognition [5]–[9], where an action video is considered as a time series of frames. Approaches [5], [6] use a so-called self-similarity matrix (SSM) descriptor to summarize actions in various views and have shown their robustness in cross-view scenarios. Information shared between views is learned and transferred to each of the views in [7]–[9]. They assume samples in different views contribute equally to the shared features. However, this assumption is not valid as the cues in one view may be remarkably different from other views (e.g., the top view in Figure 1(b)) and should have lower contribution to the shared features compared to other views. In addition, they do not constrain information sharing between action categories. This may yield similar features for videos in different classes but are captured from the same view, which would undoubtedly confuse classifiers.

We propose novel deep networks that learn view-invariant features for cross-view action classification. The action data used in this work are assumed to capture human actions. We present a novel *sample-affinity matrix* (SAM) to measure the similarities between video samples in different camera views. This allows us to accurately balance information transfer between views and help learn more informative shared features for cross-view action classification. The structure of SAM also limits information transfer between samples in different classes, which enables us to learn distinctive features in each class. In addition to the shared features, private features are also learned to capture motion information exclusively exists in each view that cannot be modeled using shared features. We separately learn discriminative view-invariant information from shared and private features by encouraging incoherence between them. The performance of the proposed

Manuscript received November 16, 2016; revised March 27, 2017; accepted April 8, 2017. Date of publication April 24, 2017; date of current version May 3, 2017. This work was supported in part by the NSF IIS under Award 1651902, in part by ONR Young Investigator under Award N00014-14-1-0484, and in part by the U.S. Army Research Office Young Investigator under Award W911NF-14-1-0218. The associate editor coordinating the review of this manuscript and approving it for publication was Mr. Pierre-Marc Jodoin. *(Corresponding author: Yu Kong.)*

Y. Kong, Z. Ding, and J. Li are with the Department of ECE, Northeastern University, Boston, MA 02115 USA (e-mail: yukong@ece.neu.edu; allanding@ece.neu.edu; junl.mldl@gmail.com).

Y. Fu is with the Department of ECE, College of Computer and Information Science, Northeastern University, Boston, MA 02115 USA (e-mail: yunfu@ece.neu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2017.2696786

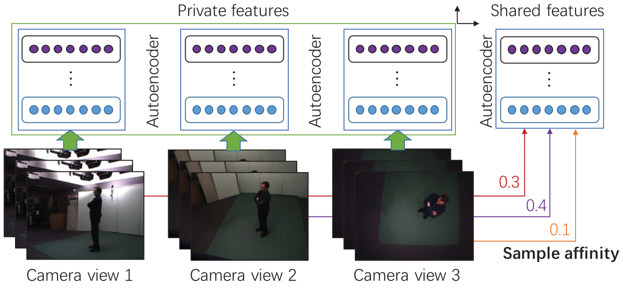


Fig. 2. Overview of the proposed method.

approach can be further boosted using label information and stacking multiple layers of features.

We formulate this feature learning problem in a marginalized autoencoder framework (see Figure 2) [10], particularly designed for learning view-invariant features. Our approaches learn two types of features, the *shared features* across views summarized by one autoencoder, as well as robust *private features* particular for one view using a group of autoencoders. Sample affinity encoded in SAM is elegantly incorporated in the learning of shared features in order to weigh different contributions of video samples. Different from the SSM [5], SAM computes sample similarity while SSM focuses on video frames. The incoherence between the two types of features is realized by encouraging the orthogonality between mapping matrices in the two categories of autoencoders. A Laplacian graph is built to encourage samples in the same action categories to have similar shared and private features. We stack multiple layers of features and learn them in a layer-wise fashion. Extensive experiments on three multi-view datasets show that our approach significantly outperforms state-of-the-art approaches.

Our contributions are threefold: 1) a new SAM is introduced to balance the contributions of samples in different views in the learning of shared features; 2) both shared and private features are learned to build robust view-invariant features; 3) extensive results show that our approach achieves remarkably higher results than existing approaches.

II. Related Work

Multi-view learning methods aim at finding mutual agreement between two distinct views of data. Extensive attempts have been made to learn more expressive and discriminative features from low-level observations [2], [11]–[16]. Co-training approach [17] trains multiple learning algorithms for each view and finds the consistent relationships between a pair of data points across different views. Canonical correlation analysis (CCA) was also used in [18] to learn a common space between multiple views. The method in [19] learns two projection matrices to map multi-modal data onto a common feature space, in which cross-modal data matching can be performed. Incomplete view problem was studied in [20]. They assumed that different views are generated from a shared subspace. A generalized multiview analysis (GMA) method was introduced in [21]. GMA is proved to be a supervised extension of CCA, and is a generalized instance of CCA, bilinear model, and partial least square. Liu *et al.* [13] used matrix factorization in multi-view clustering. Their method regularizes factors representing clustering structures learned from multiple views toward a common consensus. A collective matrix factorization (CMF) method was presented in [12], which captures correlations between relational feature matrices. Ding *et al.* [16] proposed a low-rank constrained matrix factorization model to address the multi-view learning scenario when the view information of test data is unknown.

View-invariant action recognition methods aim at predicting action labels given multi-view samples. Due to viewpoint changes, large within-class pose and appearance variation exist. Previous studies attempt to design view-invariant features that are robust to viewpoint variations. The method in [22] performs local partitioning and hierarchical classification of the 3D Histogram of Oriented Gradients (HOG) descriptor to represent sequences of images. SSM-based approaches [5], [23] compute frame-wise similarity matrix in a video and extract view-invariant descriptors within a log-polar block on the matrix. A multitask learning approach was proposed in [6] to enhance the representation power of SSM by sharing discriminative SSM features among views. Sharing knowledge between views was investigated in [7]–[9] and [24]–[28]. Specifically, MRM-Lasso method in [9] captured latent corrections across different views by learning a low-rank matrix consisting of pattern-specific weights. Transferable dictionary pairs were learned in [7] and [8], which encourage the shared feature space to be sparse. Bipartite graph was adopted in [25] to co-cluster two view-dependent vocabularies into visual-word clusters called bilingual-words in order to bridge the semantic gap across view-dependent vocabularies.

Comparisons: Different from existing multi-view learning approaches [17]–[21], [29], the proposed approach allows us to stack multiple layers of learners to learn view-invariant features in a coarse to fine fashion. Private features are also exploited in the proposed approach to capture complex motion information that uniquely exists in specific views, and are encouraged to be incoherent with shared features. Compared with knowledge sharing approaches for view-invariant action recognition [7]–[9], [24]–[26], [28], our approach balances information sharing between views based on sample similarities. This allows us to better differentiate various categories if data samples appear similar in some views. Different from [29], SAM Z directly captures with-in class between-view information and between class with-in view information, while [29] compute the between-class and within-class Laplacian matrices. SAM Z in our work measures the distance between two views of the same sample, while [29] does not encode such distance.

III. Deeply Learned View-Invariant Features

The aim of this work is to build view-invariant features that allow us to train the classification model on one (or multiple) view(s), and test on the other view.

A. Sample-Affinity Matrix (SAM)

We introduce SAM to measure the similarity between pairs of video samples in multiple views. Suppose that we are given training videos of V views: $\{X^v, \mathbf{y}^v\}_{v=1}^V$. The data of the v-th view X^v consist of N action videos: $X^v = [\mathbf{x}_1^v, \cdots, \mathbf{x}_N^v] \in \mathbb{R}^{d\times N}$ with corresponding labels $\mathbf{y}^v = [y_1^v, \cdots, y_N^v]$. SAM $Z \in \mathbb{R}^{VN\times VN}$ is defined as a block diagonal matrix:

$$Z = \mathrm{diag}(Z_1, \cdots, Z_N),\, Z_i = \begin{pmatrix} 0 & z_i^{12} & \cdots & z_i^{1V} \\ z_i^{21} & 0 & \cdots & z_i^{2V} \\ \vdots & \vdots & \vdots & \vdots \\ z_i^{V1} & z_i^{V2} & \cdots & 0 \end{pmatrix},$$

where $\mathrm{diag}(\cdot)$ creates a diagonal matrix, and z_i^{uv} is the distance between two views in the i-th sample computed by $z_i^{uv} = \exp(\|\mathbf{x}_i^v - \mathbf{x}_i^u\|^2/2c)$ parameterized by c.

Essentially, SAM Z captures within-class between-view information and between-class within-view information. A block Z_i in Z characterizes appearance variations in different views within one class. This tells us how an action varies if view changes. Such information allows us to transfer information between views and build robust cross-view features. In addition, since the off-diagonal blocks in SAM Z are zeros, it limits information sharing between classes in the same view. Consequently, the features from different classes but in the same view are encouraged to be distinct. This enables us to differentiate various action categories if they appear similarly in some views.

B. Preliminary on Autoencoders

Our approach builds upon a popular deep learning approach, Autoencoder (AE) [10], [30], [31]. AE maps the raw inputs $\mathbf{x}$ to hidden units $\mathbf{h}$ using an "encoder" $\mathbf{f}_1(\cdot)$: $\mathbf{h} = \mathbf{f}_1(\mathbf{x})$, and then maps the hidden units to outputs using a "decoder" $\mathbf{f}_2(\cdot)$: $\mathbf{o} = \mathbf{f}_2(\mathbf{h})$. The objective of learning AE is to encourage similar or identical input-output pairs where the reconstruction loss is minimized after decoding: $\min \sum_{i=1}^N \|\mathbf{x}_i - \mathbf{f}_2(\mathbf{f}_1(\mathbf{x}_i))\|^2$. Here, N is the number of training samples. In this way, the neurons in the hidden layer are good representations for the inputs as the reconstruction process captures the intrinsic structure of the input data.

As opposed to the two-level encoding and decoding in AE, marginalized stacked denoising Autoencoder [10] (mSDA) reconstructs the corrupted inputs using a single mapping W: $\min \sum_{i=1}^N \|\mathbf{x}_i - W\tilde{\mathbf{x}}_i\|^2$, where $\tilde{\mathbf{x}}_i$ is the corrupted version of $\mathbf{x}_i$ obtained by setting each feature to 0 with a probability p. mSDA performs m passes over the training set, each time with different corruptions. This essentially performs a dropout regularization on the mSDA [32]. By setting $m \to \infty$, mSDA effectively uses infinite many copies of noisy data to compute the transformation matrix W that is robust to noise. mSDA is stackable and can be solved in closed-form.

C. Single-Layer Feature Learning

The proposed model builds on mSDA. We attempt to learn both discriminative shared features between multiple views and private features particularly owned by one view for cross-view action classification. Considering large motion variations in different views, we incorporate SAM Z in learning shared features to balance information transfer between views in order to build more robust features.

We learn shared features and private features using the following objective function:

$$\min_{W,\{G^v\}} \mathcal{Q}, \quad \mathcal{Q} = \|W\tilde{X} - XZ\|_F^2 + \sum_v \Big[\alpha\|G^v\tilde{X}^v - X^v\|_F^2 + \beta\|W^{\mathrm{T}}G^v\|_F^2 + \gamma\,\mathrm{Tr}(P^vX^vLX^{v\mathrm{T}}P^{v\mathrm{T}})\Big], \tag{1}$$

where W is the mapping matrix for learning shared features, $\{G^v\}_{v=1}^V$ is a group of mapping matrices for learning private features particularly for each view, and $P^v = (W; G^v)$. The above objective function consists of 4 terms: $\psi = \|W\tilde{X} - XZ\|_F^2$ learns shared features between views, which essentially reconstructs an action data from one view using the data from all the views; $\phi_v = \|G^v\tilde{X}^v - X^v\|_F^2$ learns view-specific private features that are complementary to the shared features; $r_{1v} = \|W^{\mathrm{T}}G^v\|_F^2$ and $r_{2v} = \mathrm{Tr}(P^vX^vLX^{v\mathrm{T}}P^{v\mathrm{T}})$ are model regularizers. Here, r_{1v} reduces redundancy between two mapping matrices, and r_{2v} encourages the shared and private features of the same class and the same view to be similar. α, β, γ are parameters balancing the importance of these components. Details about these terms are discussed in the following.

Note that in cross-view action recognition, data from all the views are available in training for learning shared and private features. Data from some views are not available only in testing.

1) Shared Features: Humans can recognize an action from one view and imagine what will the action look like if we observe from other views. This is possibly because we have observed similar actions before from multiple views. This motivates us to reconstruct an action data from one view (target view) using the action data from all the views (source view). In this way, information shared between views can be summarized and transferred to the target view.

We define the discrepancy between the data of the v-th target view and the data of all the V source views as

$$\psi = \sum_{i=1}^N \sum_{v=1}^V \|W\tilde{\mathbf{x}}_i^v - \sum_u \mathbf{x}_i^u z_i^{uv}\|^2 = \|W\tilde{X} - XZ\|_F^2, \tag{2}$$

where z_i^{uv} is a weight measuring the contributions of the u-th view action in the reconstruction of the sample $\mathbf{x}_i^v$ of the v-th view. $W \in \mathbb{R}^{d\times d}$ is a single linear mapping for the corrupted input $\tilde{\mathbf{x}}_i^v$ of all the views. $Z \in \mathbb{R}^{VN\times VN}$ is a sample-affinity matrix encoding all the weights $\{z_i^{uv}\}$. Matrices $X, \tilde{X} \in \mathbb{R}^{d\times VN}$ denote the input training matrix and the corresponding corrupted version of X, respectively [10]. The corruption essentially performs a dropout regularization on the model [32].

The SAM Z here allows us to accurately balance information transfer between views and helps learn more discriminative shared features. Instead of using equal weights [7], [8], we reconstruct the i-th training sample of the v-th view using the samples from all V views with different contributions. As shown in Figure 3, a sample of side view (source 1) will be more similar to the one also from side view (target view) than the one from top view (source 2). Thus, more weight should be given to source 1 in order to learn more descriptive shared features for the target view. Note that SAM Z limits information sharing across samples (off-diagonal blocks are zeros) as it cannot capture view-invariant information for cross-view action recognition.

2) Private Features: Besides the information shared across views, there is still some remaining discriminative information that exclusively exists in each view. In order to utilize such information and make it robust to viewpoint variations, we adopt the robust feature learning in [10], and learn view-specific private features for the samples in the v-th view using a mapping matrix $G^v \in \mathbb{R}^{d \times d}$:

$$\phi_v = \sum_{i=1}^{N} \|G^v \tilde{\mathbf{x}}_i^v - \mathbf{x}_i^v\|^2 = \|G^v \tilde{X}^v - X^v\|_F^2. \tag{3}$$

Here, $\tilde{X}^v$ is the corrupted version of the feature matrix X^v of the v-th view. We will learn V mapping matrices $\{G^v\}_{v=1}^V$ given corresponding inputs of different views.

It should be noted that using Eq. (3) may also captures some redundant shared information from the v-th view. In this work, we reduce such redundancy by encouraging the incoherence between the view-shared mapping matrix W and view-specific mapping matrix G^v:

$$r_{1v} = \|W^{\mathbf{T}} G^v\|_F^2. \tag{4}$$

The incoherence between W and $\{G^v\}$ enables our approach to independently exploit the discriminative information contained in the view-specific features and view-shared features.

3) Label Information: An action data captured from various views may have large motion and posture variations. Therefore, the shared and private features extracted using Eq. (2) and Eq. (3) may not be discriminative enough for classifying actions with large variations. We address this problem by enforcing the shared and private features of the same class and same view to be similar. A within-class within-view variance is defined in order to regularize the learning of the view-shared mapping matrix W and view-specific mapping matrix G^v:

$$\begin{aligned} r_{2v} &= \sum_{i=1}^{N}\sum_{j=1}^{N}\Big[\|W\mathbf{x}_i^v - W\mathbf{x}_j^v\|^2 + \|G^v\mathbf{x}_i^v - G^v\mathbf{x}_j^v\|^2\Big] \\ &= \mathrm{Tr}(WX^v L X^{v\mathrm{T}} W^{\mathrm{T}}) + \mathrm{Tr}(G^v X^v L X^{v\mathrm{T}} G^{v\mathrm{T}}) \\ &= \mathrm{Tr}(P^v X^v L X^{v\mathrm{T}} P^{v\mathrm{T}}), \end{aligned} \tag{5}$$

Here, $L \in \mathbb{R}^{N \times N}$ is the label-view Laplacian matrix: $L = D - A$. D is the diagonal degree matrix with $D_{(i,i)} = \sum_{j=1}^{N} a_{(i,j)}$. A is the adjacent matrix that represents the label relationships of training videos. The (i, j)-th element $a_{(i,j)}$ in A is 1 if $y_i = y_j$ and 0 otherwise.

Note that we do not require features from different views in the same class to be similar as we have implicitly used this idea in Eq. (2). In learning the shared feature, features of the same class from multiple views will be mapped to a new space using the mapping matrix W. Consequently, the projected features of one sample can be better represented by the features from multiple views of the same sample. Therefore, the discrepancy between views is minimized, and thus makes within-class cross-view variance in Eq. (5) not necessary.

4) Discussion: Using label information in Eq. (5) results in a supervised approach. We can also remove this term and derive an unsupervised one by making $\gamma = 0$. We refer to the **unsupervised** approach as **Ours-1** and the **supervised** approach as **Ours-2** in the following discussions.

D. Learning

We solve the optimization problem in Eq. (1) and optimize parameters W and $\{G^v\}_{v=1}^V$ using a coordinate descent algorithm. More specifically, in each step, one parameter matrix is updated by fixing the others, and computing the derivative of $\mathcal{Q}$ w.r.t. to the parameter and setting it to 0.

1) Update W: Parameters $\{G^v\}_{v=1}^V$ are fixed in updating W. W can be updated by setting the derivative $\frac{\partial \mathcal{Q}}{\partial W} = 0$, deriving:

$$\begin{aligned} W = \Big[\sum_v (\beta G^v G^{v\mathrm{T}} + \gamma X^v L X^{v\mathrm{T}} + I)\Big]^{-1} \\ \cdot (XZ\tilde{X}^{\mathrm{T}})[\tilde{X}\tilde{X}^{\mathrm{T}} + I]^{-1}. \end{aligned} \tag{6}$$

It should be noted that $XZ\tilde{X}^{\mathrm{T}}$ and $\tilde{X}\tilde{X}^{\mathrm{T}}$ are computed by repeating the corruption $m \to \infty$ times. By the weak law of large numbers [10], $XZ\tilde{X}^{\mathrm{T}}$ and $\tilde{X}\tilde{X}^{\mathrm{T}}$ can be computed by their expectations $E_p(XZ\tilde{X}^{\mathrm{T}})$ and $E_p(\tilde{X}\tilde{X}^{\mathrm{T}})$ with the corruption probability p, respectively.

2) Update G^v: Fixing W and $\{G^u\}_{u=1,u\neq v}^V$, parameter G^v is updated by setting the derivative $\frac{\partial \mathcal{Q}}{\partial G^v} = 0$, deriving:

$$\begin{aligned} G^v = \Big(\beta W W^{\mathrm{T}} + \gamma X^v L X^{v\mathrm{T}} + I\Big)^{-1} \\ \cdot (\alpha X^v \tilde{X}^{v\mathrm{T}})[\alpha \tilde{X}^v \tilde{X}^{v\mathrm{T}} + I]^{-1} \end{aligned} \tag{7}$$

Similar to the procedure of updating W, $X^v \tilde{X}^{v\mathrm{T}}$ and $\tilde{X}^v \tilde{X}^{v\mathrm{T}}$ are computed by their expectations with corruption probability p.

3) Convergence: Our learning algorithm iteratively updates W and $\{G^v\}_{v=1}^V$. The problem in Eq. (1) can be divided into $V + 1$ subproblems, each of which is a convex problem with respect to one variable. Therefore, by solving the subproblems alternatively, the learning algorithm will guarantee to find an optimal solution to each subproblem. Therefore, the algorithm will converge to a local solution.

E. Deep Architecture

Inspired by the deep architecture in [10] and [33], we also design a deep model by stacking multiple layers of feature learners proposed in Section III-C. A nonlinear feature mapping is performed layer by layer. More specifically, a nonlinear squashing function $\sigma(\cdot)$ is applied on the output of one layer: $H_w = \sigma(WX)$ and $H_g^v = \sigma(G^v X^v)$, resulting in a series of hidden feature matrices.

A layer-wise training scheme is used in this work to train the networks $\{W_k\}_{k=1}^K$, $\{G_k^v\}_{k=1,v=1}^{K,V}$ with K layers. Specifically, the outputs of the f-th layer H_{kw} and H_{kg}^v are used as the input to the $(k+1)$-th layer. The mapping matrices W_{k+1} and $\{G_{k+1}^v\}_{v=1}^V$ are then trained using these inputs. For the first layer, the inputs H_{0w} and H_{0g}^v are the raw features X and X^v, respectively. More details are shown in **Algorithm 1**.

Algorithm 1 Learning view-invariant features

```
Input: ${(\mathbf{x}_i^v, y_i)}_{i=1,v=1}^{N,V}$.
Output: $\{W_k\}_{k=1}^K$,$\{G_k^v\}_{k=1,v=1}^{K,V}$.
for Layer $k = 1$ to $K$ do
    Input $H_{(k-1)w}$ for learning $W_k$.
    Input $H_{(k-1)g}^v$ for learning $G_k^v$.
    while not converge do
        Update $W_k$ using Eq. (6);
        Update $\{G_k^v\}_{v=1}^V$ using Eq. (7);
    end while
    Compute $H_{kw}$ by: $H_{kw} = \sigma(W_k H_{(k-1)w})$.
    Compute $\{H_{kg}^v\}_{v=1}^V$ by: $H_{kg}^v = \sigma(G_k^v H_{(k-1)g}^v)$.
end for
```

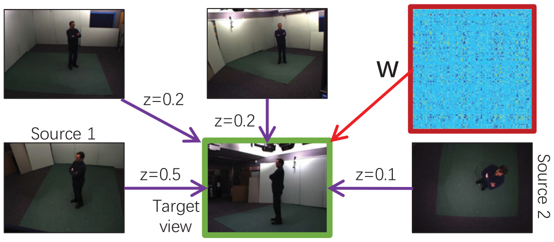


Fig. 3. Learning shared features using weighted samples.

IV. Experiments

We evaluate Ours-1 and Ours-2 approaches on three multi-view datasets: multi-view IXMAS dataset [34], Northwestern-UCLA Multiview Action 3D (NUMA) dataset [35], and the Daily and Sports Activities (DSA) dataset [1], all of which have been popularly used in [1], [7]-[9], [24], and [25].

We consider two cross-view classification scenarios in this work, Many-to-One and One-to-One. The former one trains on $V-1$ views and tests on remaining one view, while the latter one trains on one view and tests on the other view. For the v-th view that is used for testing, we simply set the corresponding X^v used in training to $\mathbf{0}$ in Eq. (1) during training. Intersection kernel support vector machine (IKSVM) with parameter $C = 1$ is adopted as the classifier. Default parameters are $\alpha = 1, \beta = 1, \gamma = 0, K = 1, p = 0$ for Ours-1 approach, and $\alpha = 1, \beta = 1, \gamma = 1, K = 1, p = 0$ for Ours-2 approach unless specified. The default number of layers is set to 1 for efficiency consideration.

IXMAS and NUMA are multi-camera-view video datasets, where each view corresponds to a camera view (see Figure 4(b) and (c)). The IXMAS dataset consists of 12 actions performed by 10 actors. An action was recorded by 4 side view cameras and 1 top view camera. Each actor repeated one action 3 times. NUMA dataset consists of 10 human actions captured by 3 Kinect sensors in 5 environments.

We adopt the *bag-of-words* model in [36]. An action video is described by a set of detected local spatiotemporal trajectory-based and global frame-based descriptors [37]. A k-means clustering method is employed to quantize these descriptors and build so-called *video words*. Consequently, a video can be represented by a histogram of the video words detected in the video, which is essentially a feature vector. An action captured by V camera views is represented by V feature vectors, each of which is a feature representation for one camera view.

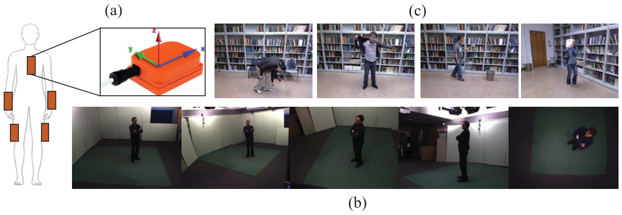


Fig. 4. Examples of multi-view problem settings: (a) multiple sensor views in the Daily and Sports Activities (DSA) dataset, and (b) and (c) multiple camera views in the IXMAS and Northwestern-UCLA datasets.

DSA is a multi-sensor-view dataset comprised of 19 daily and sports activities (e.g., sitting, playing basketball, and running on a treadmill with a speed of 8 km/h), each performed by 8 subjects in their own style for 5 minutes. 5 Xsens MTx sensor units are used on the torso, arms, and legs (Figure 4(a)), resulting in a 5-view data representation. Sensor units are calibrated to acquire data at 25 Hz sampling frequency. The 5-min signals are divided into 5-second segments so that 480(= 60seconds × 8subjects) signal segments are obtained for each activity. One 5-second segment is used as an action time series in this work.

We follow [1] to preprocess the raw action data in a 5-s window, and represent the data as a 234-dimensional feature vector. Specifically, the raw action data is represented as a 125 × 9 matrix, where 125 is the number of sampling points (125=25Hz × 5s), and 9 is the number of values (the x,y,z axes' acceleration, the x,y,z axes' rate of turn, and the x,y,z axes' Earth's magnetic field) obtained on one sensor. We first compute the minimum and maximum values, the mean, skewness, and kurtosis on the data matrix. The resulting features are concatenated and generate a 45-dimensional (5 features × 9 axes) feature vector. Then, we compute discrete Fourier transform on the raw data matrix, and select the maximum 5 Fourier peaks. This yields a 45-dimensional (5 peaks × 9 axes) feature vector. The 45 frequency values that correspond to these Fourier peaks are also extracted, resulting in a 45-dimensional (5 frequency × 9 axes) as well. Afterwards, 11 autocorrelation samples are computed for each of the 9 axes, resulting in a 99-dimensional (11 samples × 9 axes) features. The three types of features are concatenated and generate a 234-dimensional feature vector, representing the human motion captured by one sensor in a 5-second window. A human action captured by V sensors is represented by V feature vectors, each of which corresponds to a sensor view.

A. IXMAS Dataset

Dense trajectory and histogram of oriented optical flow [37] are extracted from videos. A dictionary of size 2000 is built for each type of features using k-means. We use the bag-of-words model to encode these features, and represent each video as a feature vector.

We adopt the same leave-one-action-class-out training scheme in [7], [8], and [25] for fair comparison. At each

TABLE I

ONE-TO-ONE CROSS-VIEW RECOGNITION RESULTS OF VARIOUS SUPERVISED APPROACHES ON IXMAS DATASET. EACH ROW CORRESPONDS TO A TRAINING VIEW (FROM VIEW C0 TO VIEW C4) AND EACH COLUMN IS A TEST VIEW (ALSO FROM VIEW C0 TO VIEW C4). THE RESULTS IN BRACKETS ARE THE RECOGNITION ACCURACIES OF [7], [8], AND [38] AND OUR SUPERVISED APPROACH, RESPECTIVELY

	C0	C1	C2	C3	C4
C0	NA	$(79, 98.8, 98.5, \mathbf{100})$	$(79, 99.1, \mathbf{99.7}, \mathbf{99.7})$	$(68, 99.4, 99.7, \mathbf{100})$	$(76, 92.7, 99.7, \mathbf{100})$
C1	$(72, 98.8, \mathbf{100}, \mathbf{100})$	NA	$(74, \mathbf{99.7}, 97.0, \mathbf{99.7})$	$(70, 92.7, 89.7, \mathbf{100})$	$(66, 90.6, \mathbf{100}, 99.7)$
C2	$(71, 99.4, 99.1, \mathbf{100})$	$(82, 96.4, 99.3, \mathbf{100})$	NA	$(76, 97.3, \mathbf{100}, \mathbf{100})$	$(72, 95.5, 99.7, \mathbf{100})$
C3	$(75, 98.2, 90.0, \mathbf{100})$	$(75, 97.6, 99.7, \mathbf{100})$	$(73, \mathbf{99.7}, 98.2, 99.4)$	NA	$(76, 90.0, 96.4, \mathbf{100})$
C4	$(80, 85.8, 99.7, \mathbf{100})$	$(77, 81.5, 98.3, \mathbf{100})$	$(73, 93.3, 97.0, \mathbf{100})$	$(72, 83.9, 98.9, \mathbf{100})$	NA
Ave.	$(74, 95.5, 97.2, \mathbf{100})$	$(77, 93.6, 98.3, \mathbf{100})$	$(76, 98.0, 98.7, \mathbf{99.7})$	$(73, 93.3, 97.0, \mathbf{100})$	$(72, 92.4, 98.9, \mathbf{99.9})$

TABLE II

ONE-TO-ONE CROSS-VIEW RECOGNITION RESULTS OF VARIOUS UNSUPERVISED APPROACHES ON IXMAS DATASET. EACH ROW CORRESPONDS TO A TRAINING VIEW (FROM VIEW C0 TO VIEW C4) AND EACH COLUMN IS A TEST VIEW (ALSO FROM VIEW C0 TO VIEW C4). THE RESULTS IN BRACKETS ARE THE RECOGNITION ACCURACIES OF [7], [8], [24], [25], AND [39] AND OUR UNSUPERVISED APPROACH, RESPECTIVELY

	C0	C1	C2	C3	C4
C0	NA	$(79.9, 96.7, 99.1, 92.7, 94.8, \mathbf{99.7})$	$(76.8, 97.9, 90.9, 84.2, 69.1, \mathbf{99.7})$	$(76.8, 97.6, 88.7, 83.9, 83.9, \mathbf{98.9})$	$(74.8, 84.9, 95.5, 44.2, 39.1, \mathbf{99.4})$
C1	$(81.2, 97.3, 97.8, 95.5, 90.6, \mathbf{100})$	NA	$(75.8, 96.4, 91.2, 77.6, 79.7, \mathbf{99.7})$	$(78.0, 89.7, 78.4, 86.1, 79.1, \mathbf{99.4})$	$(70.4, 81.2, 88.4, 40.9, 30.6, \mathbf{99.7})$
C2	$(79.6, 92.1, 99.4, 82.4, 72.1, \mathbf{100})$	$(76.6, 89.7, 97.6, 79.4, 86.1, \mathbf{99.7})$	NA	$(79.8, 94.9, 91.2, 85.8, 77.3, \mathbf{100})$	$(72.8, 89.1, \mathbf{100}, 71.5, 62.7, 99.7)$
C3	$(73.0, 97.0, 87.6, 82.4, 82.4, \mathbf{100})$	$(74.1, 94.2, 98.2, 80.9, 79.7, \mathbf{100})$	$(74.0, 96.7, 99.4, 82.7, 70.9, \mathbf{100})$	NA	$(66.9, 83.9, 95.4, 44.2, 37.9, \mathbf{100})$
C4	$(82.0, 83.0, 87.3, 57.1, 48.8, \mathbf{99.7})$	$(68.3, 70.6, 87.8, 48.5, 40.9, \mathbf{100})$	$(74.0, 89.7, 92.1, 78.8, 70.3, \mathbf{100})$	$(71.1, 83.7, 90.0, 51.2, 49.4, \mathbf{100})$	NA
Ave	$(79.0, 94.4, 93.0, 79.4, 74.5, \mathbf{99.9})$	$(74.7, 87.8, 95.6, 75.4, 75.4, \mathbf{99.9})$	$(75.2, 95.1, 93.4, 80.8, 72.5, \mathbf{99.9})$	$(76.4, 91.2, 87.1, 76.8, 72.4, \mathbf{99.9})$	$(71.2, 84.8, 95.1, 50.2, 42.6, \mathbf{99.7})$

time, one action class is used for testing. In order to evaluate the effectiveness of the information transfer of the proposed approaches, all the videos in this action are excluded from the feature learning procedure including k-means and the proposed approaches. Note that these videos can be seen in training the action classifiers. We evaluate both the proposed unsupervised approach (**Ours-1**) and the supervised approach (**Ours-2**).

1) One-to-One Cross-View Action Recognition: This experiment trains on data from one camera view (training view), and tests the on data from the other view (test view). We only use the learned shared features and discard the private features in this experiment as the private features learned on one view does not capture too much information of the other view.

We compare Ours-2 approach with [7], [8], and [38] and report recognition results in Table I. Ours-2 achieves the best performance in 18 out of 20 combinations, significantly better than all the comparison approaches. It should be noted that Ours-2 achieves 100% in 16 cases, demonstrating the effectiveness of the learned shared features. Thanks to the abundant discriminative information from the learned shared features and label information, our approach is robust to viewpoint variations and can achieve high performance in cross-view recognition.

We also compare Ours-1 approach with [7], [8], [24], [25], and [39], and report comparison results in Table II. Our approach achieves the best performance in 19 out of 20 combinations. In some cases, our approach outperforms the comparison approaches by a large margin, for example, C4→ C0 (C4 is the training view and C0 is the test view), C4→ C1, and C1→ C3. The overall performance of Ours-1 is slightly worse than Ours-2 due to the removal of the label information.

2) Many-to-One Cross-View Action Recognition: In this experiment, one view is used as test view and all the other views are used as training views. We evaluate the performance of our approaches in this experiment, which use both the learned shared and private features.

TABLE III

MANY-TO-ONE CROSS-VIEW ACTION RECOGNITION RESULTS ON IXMAS DATASET. EACH COLUMN CORRESPONDS TO A TEST VIEW

Methods	C0	C1	C2	C3	C4
Junejo *et al.* [5]	74.8	74.5	74.8	70.6	61.2
Liu and Shah [40]	76.7	73.3	72.0	73.0	N/A
Weinland *et al.* [22]	86.7	89.9	86.4	87.6	66.4
Liu *et al.* [25]	86.6	81.1	80.1	83.6	82.8
Zheng *et al.* [7]	98.5	99.1	99.1	100	90.3
Zheng and Jiang [8]-1	97.0	99.7	97.2	98.0	97.3
Zheng and Jiang [8]-2	99.7	99.7	98.8	99.4	99.1
Yan *et al.* [6]	91.2	87.7	82.1	81.5	79.1
No-SAM	95.3	93.9	95.3	93.1	94.7
No-private	98.6	98.1	98.3	99.4	100
No-incoherence	98.3	97.5	98.9	98.1	100
Ours-1 (unsupervised)	100	99.7	100	100	99.4
Ours-2 (supervised)	100	100	100	100	100

Our unsupervised (Ours-1) and supervised (Ours-2) approaches are compared with existing approaches [5]–[8], [22], [25], [40]. The importances of SAM Z in Eq. (2), the incoherence in Eq. (4) and the private features in Ours-2 model are also evaluated.

Table III shows that our supervised approach (Ours-2) achieves an impressive 100% recognition accuracy in all the 5 cases, and Ours-1 achieves an overall accuracy of 99.8%. Ours-1 and Ours-2 achieve superior overall performance over all the other comparison approaches, demonstrating the benefit of using both shared and private features in this work. Our approaches use the sample-affinity matrix to measure the similarities between video samples across camera views. Consequently, the learned shared features accurately characterize the commonness across views. In addition, the redundancy is reduced between shared and private features, making the learned private features more informative for classification. Although the two methods in [8] exploit private features as well, they do not measure different contributions of samples in learning the shared dictionary, making the shared information less discriminative.

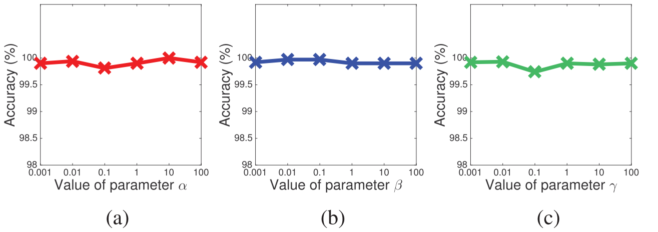


Fig. 5. Performance variations of our supervised approach (Ours-2) on IXMAS dataset with various values of parameters α, β, and γ. Note that the origin of y-axis starts from 98%. (a) Parameter α. (b) Parameter β. (c) Parameter γ

Ours-2 outperforms No-SAM approach, suggesting the effectiveness of SAM Z. Without SAM Z, No-SAM treats samples across views equally, and thus cannot accurately weigh the importance of samples in different views. The importance of the private features can be clearly seen from the performance gap between Ours-2 and No-private approach. Without private features, the No-private approach only uses shared features for classification, which are not discriminative enough if some informative motion patterns exclusively exist in one view and are not sharable across views. The performance variation between Ours-2 and the No-incoherence method suggests the benefit of encouraging the incoherence in Eq. (4). Using Eq. (4) allows us to reduce the redundancy between shared and private features, and help extract discriminative information in each of them. Ours-2 slightly outperforms Ours-1 in this experiment, indicating the effectiveness of using label information in Eq. (5).

3) Parameter Analysis: The sensitivity of our approach to parameters $\alpha, \beta, \gamma, p, K$ are evaluated in this experiment. The average performance of one-to-one cross-view action recognition accuracy is reported.

Performance variations of Ours-2 given parameters α, β, γ of values $0.001, 0.01, 0.1, 1, 10, 100$ are shown in Figure 5. Results show that our approach is insensitive to all these parameters. The largest performance gap given different parameter value of α, γ is 0.19%. The performance variation given β is even lower, which is 0.07%. These results demonstrate the insensitivity of our approach to these parameters, and thus we simply set all these parameters to 1 throughout the experiments. The results of Ours-1 are not given here as it shows similar results to Ours-2.

We also verify the effectiveness of corruption probability p in Ours-2, and the number of layers K in Ours-1 and Ours-2. Results in Figure 6 indicate that the performance slightly decreases if we increase the corruption probability p. The performance variation is only 0.68% if $p \leqslant 0.2$, and it increases to 1.25 if $p \leqslant 0.4$. The underlying reason is adding noise in raw data ($p > 0$) reduces the amount of shared information between views. Thus, the discriminative power of shared features is decreased and results in a relatively lower recognition performance. The best performance given various K is achieved at 2-layer and 3-layer in Ours-1 and Ours-2, respectively. However, the performance gap is slight, which is 0.25% and 0.82% for Ours-1 and Ours-2, respectively. Considering the extra training time using multiple layers, we use $K = 1$ in this work.

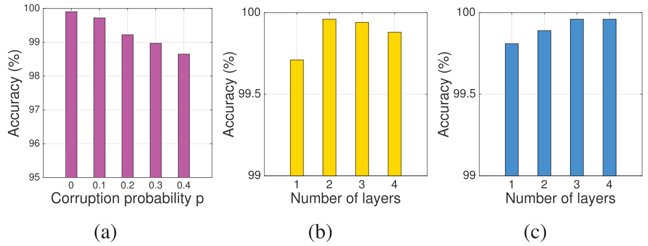


Fig. 6. Performance variations on IXMAS dataset with (left to right) different number of layers in Ours-1, and layers in Ours-2. Note that the origin of y-axis starts from 95% in (a) and 99% in (b) and (c).

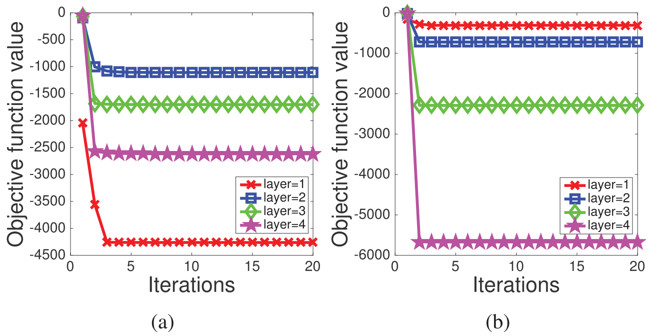


Fig. 7. Objective function values at each layer in iterations of our (a) unsupervised and (b) supervised approaches on IXMAS dataset.

The convergence of the proposed two approaches is also verified. We build 4-layer networks for the two approaches, and show the objective function values of each layer in different iterations in Figure 7. Results indicate that the training of each layer generally converges within 5 iterations.

B. Northwestern-UCLA Multiview Action 3D Dataset

1) Many-to-One Cross-View Action Recognition: We use the same features as the IXMAS dataset. Many-to-One cross-view recognition accuracy in three cross-view scenarios are reported following [35], i.e., Cross-Subject, Cross-Camera View, and Cross-Environment.

Our methods are compared with [26], [27], [35], and [41]–[43] in three cross-view scenarios following [35], i.e., Cross-Subject, Cross-Camera View, and Cross-Environment.

Results in Table IV show that Ours-2 outperforms [35]+LowR (low resolution visual features) by 3.9% and 10.4% in Cross-View and Cross-Env scenarios, respectively, and achieves comparable performance with [35]+LowR in Cross-Subject scenario. Ours-2 utilizes both private and shared features in various views, while [35] only uses features shared between views. In this comparison, the most significant performance gain of Ours-2 in Cross-Subject, Cross-View, and Cross-Env scenarios is 30.4% (over [27]), 32.0% (over [26]), and 62.3% (over [27]), respectively. Such remarkable improvements demonstrate the benefit of using both shared and private features for modeling cross-view data, and SAM for measuring the similarity of samples in multiple views. Ours-2 outperforms Ours-1 due to the use of label information.

2) Parameter Analysis: The sensitivity of our approach to parameters α, β, γ are evaluated in this experiment.

TABLE IV

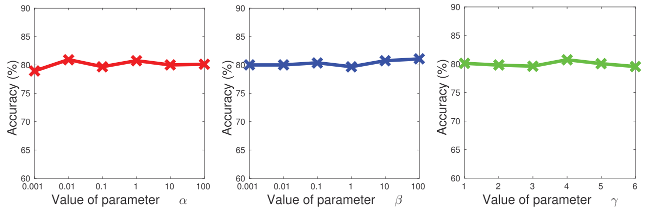
CROSS-SUBJECT, CROSS-VIEW, AND CROSS-ENVIRONMENT ACTION RECOGNITION RESULTS ON NUMA DATASET

Methods	Cross-Subject	Cross-View	Cross-Env
Li and Zickler [27]	50.7	47.8	27.4
Li *et al.* [26]	54.2	45.2	28.6
Sadanand and Corso [41]	24.6	17.6	N/A
Maji *et al.* [42]	54.9	24.5	48.5
Felzenszwalb *et al.* [43]	74.8	46.1	68.8
Wang *et al.* [35]	78.9	65.3	71.9
Wang *et al.* [35]+LowR	81.6	73.3	79.3
Ours-1 (unsupervised)	77.9	72.5	84.7
Ours-2 (supervised)	81.1	**77.2**	**89.7**

TABLE V

MANY-TO-ONE CROSS-VIEW ACTION CLASSIFICATION RESULTS ON DSA DATASET. EACH COLUMN CORRESPONDS TO A TEST VIEW. V0-V4 ARE SENSOR VIEWS ON TORSO, ARMS, AND LEGS

Methods	Overall	V0	V1	V2	V3	V4
IKSVM	54.6	36.5	53.4	63.4	60.1	59.7
DRRL [44]	55.4	35.5	56.7	62.1	61.7	60.9
mSDA [10]	56.1	34.4	57.7	62.8	61.5	64.1
No-SAM	55.4	35.1	57.0	60.7	62.2	62.2
No-private	55.4	35.1	57.0	60.7	62.2	62.1
No-incoherence	55.4	35.1	56.9	60.7	62.2	62.2
Ours-1	57.1	35.7	57.4	64.4	64.2	63.9
Ours-2	**58.0**	36.1	**58.9**	**65.8**	**64.2**	**65.2**



(a) (b) (c)

Fig. 8. Performance variations of our supervised approach (Ours-2) with various values of parameters α, β, and γ on NUMA dataset. (a) Parameter α. (b) Parameter β. (c) Parameter γ.

The average performance of many-to-one cross-view action recognition accuracy is reported.

Performance variations of Ours-2 given parameters α, β, γ of values $0.001, 0.01, 0.1, 1, 10, 100$ are shown in Figure 8. Results show that Our-2 is insensitive to all these parameters. The largest performance gap given different parameter values of α,β, and γ is within 2%. These results demonstrate the insensitivity of our approach to these parameters, and thus we simply set all these parameters to 1 throughout the experiments. The results of our unsupervised approach is not given here as it shows similar results to the supervised approach.

C. Daily and Sports Activities Data Set

1) Many-to-One Cross-View Action Classification: In this experiment, data from 4 sensors are used for training ($36,480$ time series) and the data from the remaining 1 sensor ($9,120$ time series) are used for testing. This process is repeated 5 times and the average results are reported.

Our unsupervised (**Ours-1**) and supervised (**Ours-2**) approaches are compared with mSDA [10], DRRL [44] and IKSVM. The importances of SAM Z in Eq. (2), the incoherence in Eq. (4) and the private features in Ours-2 model are also evaluated. We remove Z in Eq. (2) and the incoherence component in Eq. (4) from the supervised model, respectively, and obtain the "No-SAM", and the "No-incoherence" model. We also remove the learning of parameter $\{G^v\}_{v=1}^V$ from the supervised model and obtain the "No-private" model. Comparison results are shown in Table V.

Ours-2 achieves superior performance over all the other comparison methods in all the 5 cases with an overall recognition accuracy of 58.0%. Ours-2 outperforms Ours-1 by 0.9% in overall classification result due to the use of label information. Note that cross-view classification on DSA dataset is challenging as the sensors on different body parts are weakly correlated. The sensor on torso (V0) has the weakest correlations with the other four sensors on arms and legs. Therefore, results of all the approaches on V0 are the lowest performance compared to sensors V1-V4. Ours-1 and Ours-2 achieve superior overall performance over the comparison approaches IKSVM and mSDA due to the use of both shared and private features. IKSVM and mSDA do not discover shared and private features, and thus cannot use correlations between views and exclusive information in each view for classification. To better balance the information transfer between views, Ours-1 and Ours-2 use the sample-affinity matrix to measure the similarities between video samples across camera views. Thus, the learned shared features accurately characterize the commonness across views. Though the overall improvement of Ours-1 and Ours-2 over mSDA is 1% and 1.9%, Ours-1 and Ours-2 correctly classifies 456 and 866 more sequences than mSDA in this experiment, respectively.

The performance gap between Ours-2 and the No-SAM approach suggests the effectiveness of SAM Z. Without SAM Z, No-SAM treats samples across views equally, and thus cannot accurately weigh the importance of samples in different views. Ours-2 outperforms No-private approach, suggesting the importance of the private features in learning discriminative features for multi-view classification. Without private features, No-private approach only uses shared features for classification, which are not discriminative enough if some informative motion patterns exclusively exist in one view and are not sharable across views. Ours-2 achieves superior performance over No-incoherence method, indicating the benefit of encouraging the incoherence in Eq. (4). Using Eq. (4) allows us to reduce the redundancy between shared and private features, and help extract discriminative information in each of them. Ours-2 slightly outperforms Ours-1, indicating the effectiveness of using label information in Eq. (5).

2) Parameter Analysis: We also evaluate the sensitivity of our approach to parameters α, β, γ. In this experiment, all the 5 views are used for both training and testing. 50% action data (regardless of data views) are used for training and the remaining 50% data are used for testing.

Figure 9 illustrates the performance variations of Ours-2 given parameters α, β, γ of values $0.001, 0.01, 0.1, 1, 10, 100$. Results show that Ours-2 is insensitive to all these parameters.

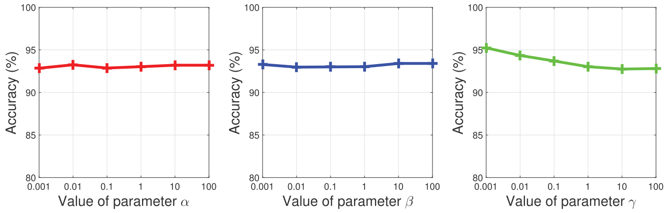
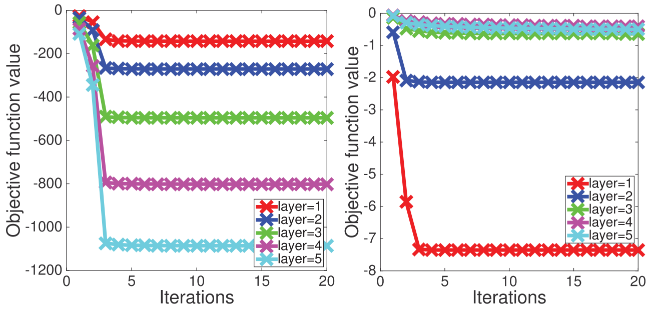


Fig. 9. Performance variations of our supervised approach (Ours-2) with various values of parameters α, β, and γ on the DSA dataset. The origin of y-axis starts from 80% in order to show slight performance variations.



(a)

(b)

Fig. 10. Objective function values at each layer in iterations of our (a) unsupervised and (b) supervised approaches on DSA dataset.

The largest performance variations given different parameter values of α, β, γ are 0.4%, 0.4%, 2.5%, respectively. The performance gap with respect to γ is slightly larger than the parameters α, β as γ determines the amount of label information to the model. Ours-2 relies on shared and private features more than the label information as it yields the best accuracy when $\gamma = 0.001$. As Ours-2 is insensitive to these parameters, we simply set all these parameters to 1 throughout the experiments. The results of Ours-1 approach are not shown here as it shows similar results to Ours-2 approach.

We also evaluate the convergence of Ours-1 and Ours-2. 5-layer networks are learned for the two approaches, and their objective function values of each layer in 20 iterations are shown in Figure 10. Results demonstrate that the training of each layer quickly converges within 5 iterations.

V. Conclusion

We have proposed two novel view-invariant feature learning approaches for cross-view action classification. Our approaches utilize both shared and private features to accurately characterize human actions with large viewpoint and appearance variations. The sample affinity matrix is introduced in this work to compute sample similarities across views. The matrix is elegantly embedded in the learning of shared features in order to accurately weigh the contribution of each sample to the shared features, and balance information transfer. Extensive experiments on the IXMAS, NUMA, and DSA datasets show that our approaches outperform state-of-the-art approaches in cross-view action classification.

References

[1] K. Altun, B. Barshan, and O. Tunçel, "Comparative study on classifying human activities with miniature inertial and magnetic sensors," *Pattern Recognit.*, vol. 43, no. 10, pp. 3605–3620, Oct. 2010.

[2] J. Grabocka, A. Nanopoulos, and L. Schmidt-Thieme, "Classification of sparse time series via supervised matrix factorization," in *Proc. AAAI*, Jul. 2012, pp. 928–934.

[3] Y. Kong and Y. Fu, "Bilinear heterogeneous information machine for RGB-D action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1054–1062.

[4] Y. Kong and Y. Fu, "Max-margin action prediction machine," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1844–1858, Sep. 2016.

[5] I. N. Junejo, E. Dexter, I. Laptev, and P. Pérez, "Cross-view action recognition from temporal self-similarities," in *Proc. ECCV*, 2008, pp 293–306.

[6] Y. Yan, E. Ricci, S. Subramanian, G. Liu, and N. Sebe, "Multitask linear discriminant analysis for view invariant action recognition," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5599–5611, Dec. 2014.

[7] J. Zheng, Z. Jiang, "Jonathon phillips and rama chellappa. Cross-view action recognition via a transferable dictionary pair," in *Proc. Brit. Mach. Vis. Conf.*, 2012, pp. 125.1–125.11.

[8] J. Zheng and Z. Jiang, "Learning view-invariant sparse representations for cross-view action recognition," in *Proc. ICCV*, Dec. 2013, pp. 3176–3183.

[9] W. Yang, Y. Gao, Y. Shi, and L. Cao, "MRM-lasso: A sparse multiview feature selection method via low-rank analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 11, pp. 2801–2815, Nov. 2015.

[10] M. Chen, Z. Xu, K. Weinberger, and F. Sha, "Marginalized denoising autoencoders for domain adaptation," in *Proc. ICML*, 2012, pp. 1627–1634.

[11] G. Ding, Y. Guo, and J. Zhou, "Collective matrix factorization hashing for multimodal data," in *Proc. CVPR*, Jun. 2014, pp. 2075–2082.

[12] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proc. KDD*, Aug. 2008, pp. 650–658.

[13] J. Liu, C. Wang, J. Gao, and J. Han, "Multi-view clustering via joint nonnegative matrix factorization," in *Proc. SDM*, May 2013, pp. 252–260.

[14] L. Liu and L. Shao, "Learning discriminative representations from RGB-D video data," in *Proc. IJCAI*, Aug. 2013, pp. 1493–1500.

[15] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Mach. Learn.*, vol. 73, no. 3, pp. 243–272, 2008.

[16] Z. Ding and Y. Fu, "Low-rank common subspace for multi-view learning," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2014, pp. 110–119.

[17] A. Kumar and H. Daumé, "A co-training approach for multi-view spectral clustering," in *Proc. ICML*, 2011, pp. 393–400.

[18] W. Zhang, K. Zhang, P. Gu, and X. Xue, "Multi-view embedding learning for incompletely labeled data," in *Proc. IJCAI*, Jun. 2013, pp. 1910–1916.

[19] K. Wang, R. He, W. Wang, L. Wang, and T. Tan, "Learning coupled feature spaces for cross-modal matching," in *Proc. ICCV*, Dec. 2013. pp. 2088–2095.

[20] C. Xu, D. Tao, and C. Xu, "Multi-view learning with incomplete views," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5812–5825, Dec. 2015.

[21] A. Sharma, A. Kumar, H. Daume, III, and D. W. Jacobs, "Generalized multiview analysis: A discriminative latent space," in *Proc. CVPR*, Jun. 2012, pp. 2160–2167.

[22] D. Weinland, M. Özuysal, and P. Fua, "Making action recognition robust to occlusions and viewpoint changes," in *Proc. ECCV*, Sep. 2010, pp. 635–648.

[23] I. N. Junejo, E. Dexter, I. Laptev, and P. Perez, "View-independent action recognition from temporal self-similarities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 172–185, Jan. 2011.

[24] H. Rahmani and A. Mian, "Learning a non-linear knowledge transfer model for cross-view action recognition," in *Proc. CVPR*, Jun. 2015, pp. 2458–2466.

[25] J. Liu, M. Shah, B. Kuipers, and S. Savarese, "Cross-view action recognition via view knowledge transfer," in *Proc. CVPR*, Jun. 2011, pp. 3209–3216.

[26] B. Li, O. I. Camps, and M. Sznaier, "Cross-view activity recognition using Hankelets," in *Proc. CVPR*, Jun. 2012, pp. 1362–1369.

[27] R. Li and T. Zickler, "Discriminative virtual views for cross-view action recognition," in *Proc. CVPR*, Jun. 2012, pp. 2855–2862.

[28] Z. Zhang, C. Wang, B. Xiao, W. Zhou, S. Liu, and C. Shi, "Cross-view action recognition via a continuous virtual path," in *Proc. CVPR*, Jun. 2013, pp. 2690–2697.

[29] M. Kan, S. Shan, H. Zhang, S. Lao, and X. Chen, "Multi-view discriminant analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 188–194, Jan. 2016.

[30] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[31] J. Li, T. Zhang, W. Luo, J. Yang, X. Yuan, and J. Zhang, "Sparseness analysis in the pretraining of deep neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2016.2541681.

[32] M. Chen, K. Weinberger, F. Sha, and Y. Bengio, "Marginalized denoising auto-encoders for nonlinear representations," in *Proc. ICML*, Jun. 2014, pp. 1476–1484.

[33] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, Dec. 2010.

[34] D. Weinland, R. Ronfard, and E. Boyer, "Free viewpoint action recognition using motion history volumes," *Comput. Vis. Image Understand.*, vol. 104, nos. 2–3, pp. 249–257, Dec. 2006.

[35] J. Wang, X. Nie, Y. Xia, Y. Wu, and S.-C. Zhu, "Cross-view action modeling, learning and recognition," in *Proc. CVPR*, Jun. 2014, pp. 2649–2656.

[36] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Proc. VS-PETS*, Oct. 2005, pp. 65–72.

[37] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *Int. J. Comput. Vis.*, vol. 103, no. 1, pp. 60–79, 2013.

[38] A. Farhadi, M. K. Tabrizi, I. Endres, and D. A. Forsyth, "A latent model of discriminative aspect," in *Proc. ICCV*, Oct. 2009, pp. 948–955.

[39] A. Gupta, J. Martinez, J. J. Little, and R. J. Woodham, "3D pose from motion for cross-view action recognition via non-linear circulant temporal encoding," in *Proc. CVPR*, Jun. 2014, pp. 2601–2608.

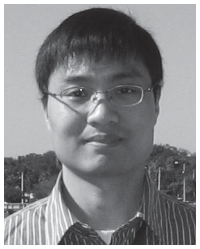
[40] J. Liu and M. Shah, "Learning human actions via information maximizationn," in *Proc. CVPR*, Jun. 2008, pp. 1–8.

[41] S. Sadanand and J. J. Corso, "Action bank: a high-level representation of activity in video," in *Proc. CVPR*, Jun. 2012, pp. 1234–1241.

[42] S. Maji, L. Bourdev, and J. Malik, "Action recognition from a distributed representation of pose and appearance," in *Proc. CVPR*, Jun. 2011, pp. 3177–3184.

[43] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[44] Y. Kong and Y. Fu, "Discriminative relational representation learning for RGB-D action recognition," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2856–2865, Jun. 2016.



Yu Kong (M'15) received the B.Eng. degree in automation from Anhui University in 2006, and the Ph.D. degree in computer science from the Beijing Institute of Technology, China, in 2012. He was a Visiting Student with the National Laboratory of Pattern Recognition, Chinese Academy of Science, from 2007 to 2009, and a Visiting Scholar with the Department of Computer Science and Engineering, State University of New York, Buffalo, in 2012. He is currently a Post-Doctoral Research Associate with the Electrical and Computer Engineering, Northeastern University, Boston, MA, USA. His research interests include computer vision, social media analytics, and machine learning.

Zhengming Ding (S'14) received the B.Eng. degree in information security and the M.Eng. degree in computer software and theory from the University of Electronic Science and Technology of China, China, in 2010 and 2013, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Northeastern University, USA. His current research interests include machine learning and computer vision. Specifically, he devotes himself to develop scalable algorithms for challenging problems in transfer learning scenario. He is an AAAI Student Member. He was a recipient of the Student Travel Grant of the ACM MM 14, the ICDM 14, the AAAI 16, and the IJCAI 16. He received the National Institute of Justice Fellowship. He was a recipient of the best paper award (SPIE). He has served as a Reviewer of the IEEE journals, such as the IEEE Transactions on Neural Networks and Learning Systems and the IEEE Transactions on Pattern Analysis and Machine Intelligence.

Jun Li (M'16) received the B.A. degree in applied mathematics from Pan Zhi Hua University in 2006, the M.S. degree in computer application from China West Normal University in 2009, and the Ph.D. degree in pattern recognition and intelligence systems from the Nanjing University of Science and Technology in 2015. From 2012 to 2013, he was a Visiting Student with the Department of Statistics, Rutgers University, Piscataway, NJ, USA. He is currently a Post-Doctoral Research Associate with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA. His current research interests include deep learning, sparse representations, subspace clustering, and recurrent neural networks. He has authored over 20 papers in the AAAI, the IEEE TNNLS, the IEEE TIP, and other venues. He has served as a PC member of the AAAI 2017, the IJCAI 2017, the IEEE FG 2017, and the IEEE ICMLA 2016, and a Reviewer for over ten international journals, such as the IEEE TNNLS and the IEEE TIP.

Yun Fu (S'07–M'08–SM'11) received the B.Eng. degree in information engineering and the M.Eng. degree in pattern recognition and intelligence systems from Xi'an Jiaotong University, China, and the M.S. degree in statistics and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana–Champaign. He has been an Interdisciplinary Faculty Member affiliated with the College of Engineering and the College of Computer and Information Science, Northeastern University, since 2012. His research interests are machine learning, computational intelligence, big data mining, computer vision, pattern recognition, and cyber-physical systems. He has extensive publications in leading journals, books/book chapters, and international conferences/workshops. He is a fellow of the IAPR, a Lifetime Senior Member of the ACM and the SPIE, a Lifetime Member of the AAAI, the OSA, and the Institute of Mathematical Statistics, a member of the Global Young Academy and the INNS, and a Beckman Graduate Fellow from 2007 to 2008. He received seven Prestigious Young Investigator Awards from the NAE, the ONR, the ARO, the IEEE, the INNS, the UIUC, and the Grainger Foundation, seven Best Paper Awards from the IEEE, the IAPR, the SPIE, and the SIAM, three major Industrial Research Awards from Google, Samsung, and Adobe. He serves as an Associate Editor, the Chair, a PC member, and a Reviewer of many top journals and international conferences/ workshops. He is currently an Associate Editor of the IEEE Transactions on Neural Networks and Leaning Systems.