Multimedia Tools and Applications (2022) 81:28479–28507
https://doi.org/10.1007/s11042-022-12822-2

An improved open-view human action recognition with unsupervised domain adaptation

M. S. Rizal Samsudin [1] · Syed A. R. Abu-Bakar [1] · Musa M. Mokji [1]

Received: 20 May 2021 / Revised: 3 February 2022 / Accepted: 9 March 2022 /
Published online: 30 March 2022

Abstract
One of the primary concerns with open-view human action recognition (HAR) is the large differences between data distributions of the target and source views. Subsequently, such differences cause the data shift problem to occur, and hence, decreasing the performance of the system. This problem comes from the fact that real-world situation deals with unconstrained rather than constrained situations such as differences in camera resolutions, field of views, and non-uniform illumination which are not found in constrained datasets. The primary goal of this paper is to improve this open-view HAR by proposing the unsupervised domain adaptation approach. In particular, we demonstrated that the balanced weighted unified discriminant and distribution alignment (BW-UDDA) managed to handle the dataset with significant differences across views such as those found in the MCAD dataset. We showed that by using the MCAD dataset on two types of cross-view evaluations, our proposed technique outperformed other unsupervised domain adaptation methods with average accuracies of 13.38% and 61.45%. Additionally, we applied our method to a constrained multi-view IXMAS dataset and achieved an average accuracy of 90.91%. The results confirmed the superiority of the proposed technique.

Keywords Open-view · Human action recognition · Domain adaptation

✉ M. S. Rizal Samsudin
ms.rizal1986@graduate.utm.my

Syed A. R. Abu-Bakar
syed@fke.utm.my

Musa M. Mokji
musa@fke.utm.my

[1] School of Electrical Engineering, Faculty of Engineering, Universiti Teknologi Malaysia, Johor, Malaysia

1 Introduction

In recent years, the interest in human action recognition (HAR) has received substantial attention from researchers around the globe due to its wide applications. According to J. K. Aggarwal, et al. [1], an action can be interpreted as an activity involving a person with several movements. Examples of basic human actions include 'walking', 'running', 'waving', and 'punching'. These are the most commonly used basic actions to be recognized in intelligent video security surveillance purposes. It is known that HAR faces several challenges and issues such as background complexity, inter and intra-class variations, noise, occlusions, low resolution, real-time processing, and view invariance.

Previously, most researchers focused on single-view approaches and they have achieved remarkable results [16, 30, 35, 36]. However, methods implemented in a single-view setup cannot accurately recognize actions performed from a different viewpoint. Since then, researchers have put more effort into recognizing human action in multiple cameras setup or multi-view to achieve high view invariance performance. Although HAR based on multiple-view methods has been an active research field in the past decade [12, 13, 20, 26], the datasets used were limited only to a studio or constrained environment. The limitation of the constrained datasets has led to the current studies of multi-view for unconstrained datasets to reflect the real-world challenges. In studying the unconstrained multi-view HAR, two crucial issues need to be addressed: (1) *Label data*. In a real-world implementation, labels for classifying actions are simply not available. This is in contrast to the controlled dataset where labels are available and sufficient for both training and testing. (2) *Distribution shift*. Constrained datasets are typically designed to have fixed parameters such as camera pixel, resolution, illumination, field of view, simultaneously recorded and uniform background scenes. Obviously, with these constraints, the focus is to increase the performance accuracy of the action classes. However, any changes from these set parameters will certainly cause the dynamics of the features to change, and hence, cause a data shift or distribution mismatch that will result in degraded performance. Therefore, it is essential to ensure that the data is optimally discriminative before feeding into the standard classifiers.

In this paper, we conduct a study on an open-view HAR with respect to the unconstrained dataset. Previous works in multi-camera HAR were limited to applications using constrained multi-camera datasets that were constrained with the above limitations. In such a constrained arrangement, many methods have reached a near perfection performance. Some of these constrained datasets include KTH [16] and Weizmann [10] for a single camera, and IXMAS [41], WVU [15], and MUHAVI [32] for multi-cameras. Conversely, this paper focuses on handling open-view HAR cases that have been highlighted by [33] with the following characteristics:

(1) Applicable only for multi-camera datasets or within cameras.
(2) The correlation between cameras is minimized so as the dataset resembles closely to the real-world environment. Thus, many parameters such as illumination, camera type, background scenes, and split action recorded, will be allowed to be varied.
(3) No labeled data is available in the gallery (target) view.

We acknowledge that some of the existing multi-view methods are able to tackle multi-camera issues such as the transferable dictionary learning approach [52]. However, we noticed that the transferable dictionary learning technique is learned directly from the original space of the

source and target views and needs a rigid assumption that the distribution data is identical across the view. Contrary to this, open-view HAR tends to have large differences in the data distribution, and therefore, the use of a transferable dictionary learning approach is unsuitable to handle this issue.

To alleviate the different data distribution issues with limited or even missing labels, unsupervised domain adaptation methods have been proposed [29]. The purpose of the unsupervised domain adaptation is to reduce the distribution shift in low-level dimensional spaces effectively so that previously labeled source view feature data can be used in target view feature data. Inspired by their achievements, this paper initially, analyzes the effectiveness of unsupervised domain adaptation methods in addressing the open-view HAR. Then, we also propose a new unsupervised domain adaptation technique named Balanced Weighted Unified Discriminant and Distribution Alignment (BW-UDDA). Our technique considers some of the weaknesses of the previous methods in an effort to increase the performance accuracy. To illustrate the strength of our approach, we evaluate our proposed technique by comparing it with other state-of-the-art unsupervised domain adaptation methods. In summary, the main contributions of this paper are as follow:

(1) We extend the study of HAR onto an open-view platform. Previous studies on HAR concentrate on a closed-view that is limited and considered solved.

(2) We propose the use of an unsupervised domain adaptation approach to address the open-view HAR problem. To the best of our knowledge, this is the first attempt.

(3) We offer a new unsupervised domain adaptation technique leveraging on weaknesses of previous methods to boost the accuracy performance of open-view HAR.

The rest of the paper is organized as follows: Section 2 reviews all relevant work while Section 4 discusses our proposed methods. Next, Section 5 describes the experimental setups, results, and discussion, and lastly, Section 5 concludes this paper.

2 Related works

2.1 View-invariant human action recognition

From the literature, the work in view-invariant HAR can be divided into two categories: (1) *feature invariant* and (2) *knowledge transfer*.

The *feature invariant* aims to exploit any shared features between views and build a descriptor from these shared features. Junejo et al. [11] proposed a descriptor by building upon a self-similarities matrix (SSM) from the low-level features between two views. The pattern of the SSM is evaluated to observe the similarity across different viewpoints. This technique, however, is sensitive when the appearance changes. Yan et al. [45] improved the SSM performance by adding a multi-task linear discriminant analysis (LDA) to maximize class covariance and minimize the within-class covariance. Li et al. [3] introduced a low-level feature extraction from a dynamic system and Hankel matrix called 'Hankelet'. The hankelets between two views are compared with dissimilarity scores before representing the feature vectors as a bag of henkelets (BoHK). This method, however, has a similar drawback with the SSM in that it is sensitive to a large viewpoint change. Ciptadi et al. [6] proposed new local features by considering spatial and temporal information called a movement pattern histogram (MPH). The MPH encodes the global temporal pattern over the video based on optical flow tracking and, from the results, it is compatible with other descriptors in terms of view-invariance.

On the other hand, the *knowledge transfer* with transfer learning strategy aims at gathering and exploiting the statistical connection between the feature vectors (as input data) from the source view to the target view. The strategy is to build a view-invariant feature space from the source to the target domain and classify different classes from different viewpoints. Farhadi et al. [7] proposed a maximum margin clustering (MMC) approach to generate split-based features of the source view, and the transfer split values to the target view. Zhang et al. [49] added temporal information into the MMC to proposed a Contextual MMC (CMMC). However, both of these methods required feature-to-feature correspondence to train the classifiers, leading to a high computational cost. Liu et al. [20] generated bilingual words from a bipartite graph that bridged two different bags of visual words (BoVW) from different domains into a bag of bilingual word (BoBW) model. Li *et. al.* [17] introduced virtual views that connect a source view and a target view in a continuous virtual path. Later, Zhang et al. [50] modified this approach by keeping all the visual information on the virtual path without tuning the parameter. Zheng et al. [52] built dictionaries based on sparse coding for the source and target views in another development. These dictionaries represent every video with the same action with similar sparse representation. Alternatively, Kong et al. [14] considered a view specific with a view shared feature learning by using a marginalized stacked denoising autoencoder based on a deep learning approach.

In contrast, Wu et al. [44] proposed heterogeneous feature spaces for a source view and target view that are learned from a projection matrix. Each class is then classified by its corresponding weight. Finally, Liu et al. [21] later proposed a unified framework that incorporates [14, 52] and adapts a new distribution which they called joints sparse representation and distribution adaptation.

The view-invariant method applied in this paper falls under the category of knowledge transfer. However, different from the above works, which focus more on the closed-view dataset setting, our work concentrates on the open-view dataset setting. Our motivation is inspired by Li et al. [18] and Su et al. [33]. Li et al. [18] proposed a multi-camera action dataset (MCAD) to study the open-view HAR. However, we notice that both works have only used existing multi-view methods and are still open to many proposed methods to improve the problem.

2.2 Unsupervised domain adaptation

Domain adaptation can be viewed as a special case of transfer learning [28, 31] when the data's training and testing are not drawn from the same distribution. Consequently, a distribution mismatch is created since the classifier trained on the source domain degrades its performance when tested on the target domain. Domain adaptation methods tend to reduce the divergence between the source and the target domains until the model trained on the source performs well on the target domain. This paper focuses on the unsupervised approach of domain adaptation that is more practical for real-world applications. Nevertheless, it is very challenging for the reason that there is no label for the conditional distribution of the target domain $P(Y|X_T)$, and the fact that a discriminative model trained on the source domain $P(Y|X_S)$ cannot be leveraged. Among the possible choices then is either to utilize the marginal distribution $P(X_T)$ and $P(X_S)$, or to use a pseudo label class of the classifier to fit the target domain $P(\widehat{Y}|X_T)$. We have divided the work on unsupervised domain adaptation into two groups: (1) *data centric* and (2) *subspace centric.*

The *Data-centric* methods seek a unified or joint transformation, (.), that projects the source and target domains into a new space that reduces the discrepancy between the domains and simultaneously preserves the data properties in their original spaces [46]. Pan et al. [28] proposed the transfer component analysis (TCA) approach. This method learns by transferring components across domains that minimize the maximum mean discrepancy (MMD) between the two new representations of the two domains using a reproducing kernel Hilbert space (RKHS). Data properties are preserved in the subspace spanned by these transfer components while data discrepancies are reduced. Long et al. [24] proposed a joint distribution adaptation (JDA) approach that considers a marginal distribution shift and a conditional distribution shift using a pseudo label of the target domain. Later, the same authors also proposed a transfer joint matching (TJM) approach that adds instance reweighting into the TCA in finding the unified subspace. Ghifary et al. [9] used scatter component analysis (SCA) to convert both feature vectors in the source and target domains into scattered space in the RKHS, minimizing the divergence. Wang et al. [38] proposed a balanced distribution adaptation (BDA) approach that considers the weight to balance the importance of the marginal and conditional distributions concerning similar or dissimilar datasets. The same author then, proposed manifold dynamic distribution adaptation (MDDA) [37] which improves the BDA by manually inserting weights to the two distributions to dynamically adapt the weight difference specifically when there is a large discrepancy between the two domains.

On the other hand, the Subspace centric methods aim to manipulate the subspaces of two domains to reduce the domain shifts without directly exploiting the data (feature vectors) [46]. Fernando et al. [8] proposed a subspace alignment (SA) by aligning the source and target subspaces using a transformation matrix. This technique also considers a unified or joint transformation to reduce the discrepancy. Gong et al. [4] proposed geodesic flow kernel (GFK) which that geodesic flow kernel was used to model the domain shift by integrating an infinite number of subspaces on the geodesic flow. The trick is to explores an intrinsic low-dimensional spatial structure that associates two domains and try to find geodesic line from X_s point to X_t point so that the raw feature can be transformed in to a space of infinite dimension where distribution difference can be reduced. However, both the SA and GFK methods fail to emphasize the importance of minimizing the distributions between domains after aligning or integrating the subspaces. Due to this shortcoming, Sun et al. [34] proposed a subspace distribution alignment (SDA) to improve SA and GFK. However, their method does not align the subspaces; but instead, it aligns the source and target data distribution. Alternatively, Zhang et al. [47] proposed a joint geometrical and statistical alignment (JGSA) approach that aligns the subspace and the distribution. Compared to the SDA, JGSA does not learn the alignment matrix to map the source data to the target, instead, it simultaneously optimizes the two couple of projections from the source domain to the target domain. Wang et al. [39] proposed manifold embedded distribution alignment (MEDA) that raised the class imbalance between domains and proposed automated balanced weight factors that capitalized on the importance of statistical marginal and conditional data distribution. Apart from computing feature discrepancy by the sum of marginal and conditional distribution MMDs as [24, 25, 38, 47], Zhang et al. [48] proposed a novel theoretical basis for computing feature discrepancy through joint probability distribution discrepancy directly through the use of MMD computation. Their work is known as discriminative joint probability maximum mean discrepancy (DJP-MMD). DJP-MMD not only minimizes the discrepancy for the same class between different classes or domains but simultaneously maximizes the discrepancy between different classes or domains.

3 Formulation

This section will be divided into four subsections: (1) Current issues, (2) Problem definition, (3) low-level feature extraction and encoding, and (4) balanced weighted unified discriminant and distribution alignment.

3.1 Current issues

We have found that the best representation for the unsupervised domain adaptation approaches follows that of the data-centric model. This model aims at transforming or projecting the data into a unified or joint subspace by learning adaptation matrices that minimize the discrepancy between the distributions of the source and the target views. The standard tool to reduce the difference between domains is to use a nonparametric Maximum Mean Discrepancy (MMD) distance measure that operates in infinitely reproducing kernel Hilbert space (RKHS). However, considering only the distribution factors does not always guarantee promising results. For that reason, many previous works have combined MMD computation with dimensionality reduction methods to increase the accuracy. Among the popular dimensionality methods is the use of the unsupervised principal component analysis (PCA) [24, 25] technique. On the contrary, methods proposed in [9, 47] utilized the label information in the source domain by using linear discriminant analysis (LDA) separately for the source domain and PCA for the target view.

LDA is a supervised method used to solve binary class problems by minimizing the within-class-scatter matrix, S_w and simultaneously maximizing the between-class scatter matrix S_b. Still, the conventional LDA has the following disadvantages [5, 43]: (1) it is sensitive to outliers or noise data in square operation, and (2) it only uses global information and neglects local information. Liu et al. [23] showed that adding local information into LDA can improve the accuracy performance. Hence, this work focuses on enhancing the discriminative part in feature adaptation by adding locality-sensitive discriminant analysis (LSDA) into unsupervised domain adaptation models. LSDA uses nearest-neighbor graphs $\mathbb{G}$ with a weight matrix W that characterizes the local geometry of the data manifold to produce a within-class graph G_w and between-class graph G_b. Both the G_w and G_b in LSDA represent local information, while S_w and S_b in LDA represent global information.

We also noticed from [38] that imbalanced class often occurs in the domain adaptation paradigm. This class imbalance usually depends on the condition of the both (source and target) view: the more similar the source view and target view is, the more dominant the conditional distribution gets. On the other hand, the less similar viewpoint between source view and target view is, the more dominant the marginal distribution gets.

3.2 Problem definition

Based on the issues addressed above, we propose a joint domain adaptation framework that fuses both the local and global information to enhance performance. In addition, our proposed joint domain adaptation framework model utilizes the existing split transformation adaptation matrices [47] that reduce the divergence of the source and target views and takes advantage when the discrepancy is too large between both domains. We also utilize the classifier-based transfer and manifold regularization used in [39] to avoid feature distortion and maximize the intrinsic manifold structure of data.

Additionally, inspired by the success of [38], we extend the concept of balancing into our proposed framework. Therefore, our proposed technique consists of 3 components: (1) Subspace alignment (we adopt exactly from [47]), (2) Balance weight distribution alignment, and (3) Discriminative feature vector. The architecture for our proposed methods, BW-UDDA, is shown in Fig. 1.

To proceed with our proposed method, the following definitions will be used:

(1) *Definition 1 (View).* The source domain data, $x_s \in \mathbb{R}^{d \times m}$ is drawn from $P_s(X_s)$ and target domain $x_t \in \mathbb{R}^{d \times n}$ is drawn from $P_t(X_t)$, where d is the size of the codebook, while m and n are the sample size for x_s and x_t, respectively. Since this work focuses on unsupervised domain adaptation in open-view HAR, the source domain will be renamed as the source view and it is defined as $D_s = \{(x_i, y_i) \ldots (x_m, y_m)\}$ and the unlabelled target domain as the target view, $D_t = \{(x_j) \ldots (x_n)\}$, where $x \in \mathbb{R}^D$. We summarize the notations and symbols in Table 1.

(2) *Definition 2 (Task).* Domain adaptation deals with the dataset shift problem that makes the source distribution and the target distribution of features/labels no longer identical. What this means is that the marginal and conditional distributions of both domains are different, i.e., $P_s(X_s) \neq P_t(X_t)$, and $P_s(y_s|\ x_s) \neq P_t(y_t|\ x_t)$ even though feature space and label space for both source and target views are the same.

3.3 Low-level feature extraction and encoding

For the features, we choose the improved dense trajectories (iDTs) [43] approach which provides us with trajectory shape, histogram of oriented gradient (HOG), histogram of optical flow (HOF), and motion of boundary histogram (MBHx and MBHy) for feature extraction and encoding. We follow [21, 53] in adopting the Locality-constrained Linear Coding (LLC) [40] scheme to represent iDTs by multiple bases. The main reason is to reduce the quantization

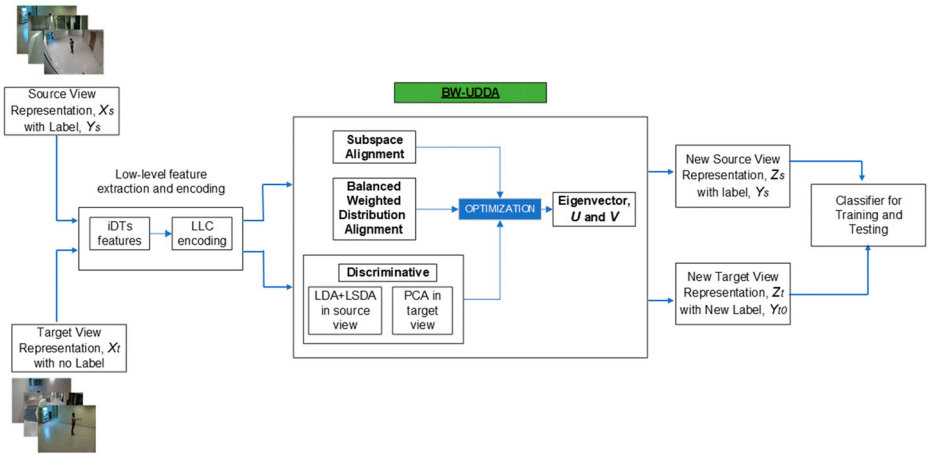


Fig. 1 Architecture of our proposed method, BW-UDDA with three components: Subspace Alignment, Balanced Weighted Distribution Alignment, and Discriminative Features Vector. Our aims were to find the adaptation matrices, U and V and then the new source and target view representation Z_s and Z_t

Table 1 Symbols and Description used in this paper

Symbols	Description	Symbols	Description
D_s, D_t	source/target view	x_s, x_t	source/target view input data
$n_s n_t$	number of samples in source/target	y_s, y_t	source/target view input labels
ω	balanced weighted factor	U, V	adaptation matrix
$\mathbb{G}$	nearest neighbor graph	$\boldsymbol{M}$	MMD matrix
S_w,S_b	within-class-scatter matrix/ between-class-scatter-matrix	W_{ij}	weight matrix of $\mathbb{G}$
$P_s(X_s)$, $P_t(X_t)$	marginal distributions source and target view	Z_s, Z_t	new representation of source/ target views
$P_s(y_s\|x_s)$, $P_t(y_t\|x_t)$	conditional distributions source and target view	$\widehat{z}$	low-dimensional data for LDA/LSDA
α, β, γ, λ, η, ζ	parameters		

error while preserving the local smooth sparsity. We follow [21] to choose the LLC scheme, and the codebook size is set to 4000 for all training-testing partitions, meaning that the encoded iDTs feature is 4000. Out of these, we only choose 200 local iDTs randomly as in [53].

3.4 Balanced weighted unified discriminant and distribution alignment (BW-UDDA)

This sub-section presents the local discriminants and introduces balanced weighted factors in feature transformation into our proposed joint domain adaptation framework. We aim to find a pair of projections, U for the source view and V for the target view, to obtain new representations Z_s and Z_t, before we feed them into the linear classifier for classification.

The BW-UDDA has four goals: (a) Adapting local discriminant into dimensionality reduction strategy, (b) Obtaining a pair of projections between the source and target views. This is achieved by minimizing and balancing the distribution divergence and subspace divergence, (c) Preserving the neighborhood structure of the dataset, and (d) Concatenating all the objectives functions and finding the optimal classifier, iteratively.

3.4.1 Dimensionality reduction

The main purpose of dimensionality reduction is to convert a high-dimensional space into a low-dimensional space so that the data can be compactly but meaningfully represented. The standard dimensionality technique used in unsupervised domain adaptation is the principal component analysis (PCA) since no class information is required. Another dimensionality reduction technique that has also been used extensively is the linear discriminant analysis (LDA) method. Not only that LDA reduces the high-dimensional space, but it also maximizes the variance between classes and minimizes the variance within a class. With these benefits, LDA is also suitable for unsupervised domain adaptation by leveraging the class information already available in the source view. However, we identified that LDA has some drawbacks, particularly it is unadaptable since it is a global discriminant and its sensitiveness to outliers and noise. Due to these limitations, we introduce locality-sensitive discriminant analysis (LSDA) along with LDA to highlight the importance of the global and local discriminants, and in that respect, we improve the accuracy performance.

First, we form the nearest neighbor graph, $\mathbb{G}$, and its weight matrix is given as follows:

$$W_{ij} = \begin{cases} 1, & x_i \in knn(x_j) \text{ or } x_j \in knn(x_i) \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $knn(x_i)$ is the set of k-NN corresponding to x_i. The obtained nearest neighbors' graph, $\mathbb{G}$ will be divided into two subgraphs: the within-class subgraph, G_w, and the between-class subgraph, G_b. Let $\widehat{Z} = (\widehat{z}_1, \widehat{z}_2, \ldots \widehat{z}_m)^{\mathrm{T}}$ be such a low-dimension data in the q-dimensional space, $\widehat{z}_i = U^T x_i$, and $\{W_w,\ W_b\}$ be the weight matrices for $\{G_w,\ G_b\}$, respectively. The objective functions can be defined as follows:

$$min \sum_{ij} \left\| \widehat{z}_i - \widehat{z}_j \right\|^2 W_{ij}^w \tag{2}$$

$$max \sum_{ij} \left\| \widehat{z}_i - \widehat{z}_j \right\|^2 W_{ij}^b \tag{3}$$

Equation (2) will incur a heavy penalty if neighboring points x_i and x_j are mapped far apart while they are actually in the same class, likewise with Eq. (3), will incur a heavy penalty if the neighboring point x_i and x_j are mapped close together while they are actually in the different classes. By minimizing Eq. (2), if x_i and x_j are close and sharing the same label, then y_i and y_j will be close as well. Likewise, by maximizing Eq. (3), if x_i and x_j are close but in a different label, then y_i and y_j will be far apart. Hence, Eq. (2) and (3) can be represented as

$$\min_U Tr\left(U^T T_w U\right) \tag{4}$$

$$\max_U Tr\left(U^T T_b U\right) \tag{5}$$

where $T_w = X_s^T L_w X_s$ and $T_b = X_s^T L_b X_s$. L_w and L_b are the Laplacian matrix of G_w and G_b, respectively. It is defined as $L_w = D^w - W^w$, $L_b = D^b - W^b$. The D^w and D^b are the diagonal matrices with diagonal entries $D_{ii}^w = \sum_j W_{ij}^w$ and $D_{ii}^b = \sum_j W_{ij}^b$, respectively. We define the global discriminant LDA as follows:

$$\min_U Tr\left(U^T S_w U\right) \tag{6}$$

$$\max_U Tr\left(U^T S_b U\right) \tag{7}$$

where $S_w = \sum_{c=1}^{C} X_s^{(c)} H_s^{(c)} \left(X_s^{(c)}\right)^T$ is within-class scatter matrix and $S_b = \sum_{c=1}^{C} n_s^{(c)} \left(\mu^{(c)} - \mu\right)\left(\mu^{(c)} - \mu\right)^T$ is the between-class scatter matrix. $H_s^{(c)}$ is the centering matrix of data within a class, $n_s^{(c)}$ is a number of class samples in class c, μ is the total sample mean vector and $\mu^{(c)}$ is the average vector belong to class c.

We unified the global and local discriminants to optimize the discriminative source information by adding Eqs. (4), (5), (6), and (7), as Eq. (8) and (9) below:

$$\min_{U} Tr\left(U^T\left(S_w+\gamma T_w\right)U\right) \tag{8}$$

$$\max_{U} Tr\left(U^T(S_b+\gamma T_b)U\right) \tag{9}$$

where γ is the balance parameter that merges global and discriminant matrices. For the target view, we only use PCA, as in Eq. (10), to maximize the variance since there is no-label information is available.

$$\max_{V} Tr\left(V^T S_t V\right) \tag{10}$$

where $S_t = X_t H_t X_t^T$ is the target view scatter matrix, while $H_t = I_t - \frac{1}{n_t} 1_t 1_t^T$ is the centering matrix.

3.4.2 Distribution and subspace divergence minimization

Unsupervised domain adaptation frameworks usually involve two transformation techniques: (a) Distribution divergence minimization, and (b) Source discriminative information preservation. We consider both methods in our joint objective function. In this sub-section, we learn two projections (both from source and target views) into respective subspaces so that the marginal and conditional distribution divergences are minimized and preserved, and the divergence of two projections is constrained to be geometrically small. We achieved this by introducing a balance factor in the process of minimizing the distribution divergence. By balancing the contribution of marginal and conditional distributions, we believe that the accuracy performance can be optimized. This is because for similar dataset, the conditional distribution becomes more dominant, and for dissimilar dataset, the marginal distribution becomes more dominant.

First, we employed maximum mean discrepancy (MMD), which computes the distance between the sample mean of source and target data in the Reproducing Kernel Hilbert Space (RKHS). The MMD computation for marginal and conditional distributions is as follows:

$$\min_{U,V} \left\| \frac{1}{n_s} \sum_{x_{si}\in \mathrm{X_s}} U^T x_{si} - \frac{1}{n_t} \sum_{x_{tj}\in \mathrm{X_t}} V^T x_{tj} \right\|_{\mathcal{H}}^2 \tag{11}$$

$$\min_{U,V} \sum_{c=1}^{C} \left\| \frac{1}{n_s^{(c)}} \sum_{x_{si}\in \mathrm{X_s}^{(c)}} U^T x_{si} - \frac{1}{n_t^{(c)}} \sum_{x_{tj}\in \mathrm{X_t}^{(c)}} V^T x_{tj} \right\|_{\mathcal{H}}^2 \tag{12}$$

From Eq. (11) and (12), we can combine the marginal and conditional distribution shift minimization to get the distribution divergence term:

$$min_{U,\mathrm{V}} \mathrm{Tr}\left(\begin{bmatrix} U^T & V^T \end{bmatrix} \begin{bmatrix} \mathrm{R_{SS}} & R_{\mathrm{st}} \\ R_{\mathrm{ts}} & R_{\mathrm{tt}} \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix} \right) \tag{13}$$

where $R_{\mathrm{ss}} = X_S \widehat{\boldsymbol{M}}_{SS} \mathrm{X}_S^T$, $R_{st} = X_S \widehat{\boldsymbol{M}}_{St} \mathrm{X}_t^T$, $R_{st} = X_t \widehat{\boldsymbol{M}}_{tS} \mathrm{X}_S^T$ and $R_{tt} = X_t \widehat{\boldsymbol{M}}_{tt} \mathrm{X}_t^T$. $\widehat{\boldsymbol{M}}$ is the MMD matrix involving marginal and conditional distributions for both marginal and conditional

distributions. We proposed to introduce a balance factor, ω inside the MMD matrix computational as follows:

$$\widehat{\boldsymbol{M}}_{\mathrm{ss}} = (1-\omega)\mathrm{M}_{\mathrm{ss}} + \omega \sum_{c=1}^{C} {M_{ss}}^{(c)}$$
$$\text{where, } M_{ss} = \frac{1}{n_s n_s} 1_s 1_s^T, \quad \left(M_{ss}^{(c)}\right)_{ij} = \begin{cases} \dfrac{1}{n_s^{(c)} n_s^{(c)}} & x_i, x_j \in \mathrm{X}_s^{(c)} \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

$$\widehat{\boldsymbol{M}}_{\mathrm{st}} = (1-\omega)\mathrm{M}_{\mathrm{st}} + \omega \sum_{c=1}^{C} {M_{st}}^{(c)}$$
$$\text{where, } M_{st} = \frac{1}{n_s n_t} 1_{\mathrm{s}} 1_t^{\mathrm{T}}, \left(M_{st}^{(c)}\right)_{ij} = \begin{cases} \dfrac{-1}{n_s^{(c)} n_t^{(c)}} & x_i \in \mathrm{X}_s^{(c)}, x_j \in \mathrm{X}_t^{(c)} \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

$$\widehat{\boldsymbol{M}}_{\mathrm{ss}} = (1-\omega)\mathrm{M}_{\mathrm{ss}} + \omega \sum_{c=1}^{C} {M_{ss}}^{(c)}$$
$$\text{where, } M_{ts} = \frac{1}{n_t n_s} 1_t 1_s^T, \left(M_{ts}^{(c)}\right)_{ij} = \begin{cases} \dfrac{-1}{n_t^{(c)} n_s^{(c)}} & x_j \in \mathrm{X}_t^{(c)}, x_i \in \mathrm{X}_s^{(c)} \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

$$\widehat{\boldsymbol{M}}_{\mathrm{ss}} = (1-\omega)\mathrm{M}_{\mathrm{ss}} + \omega \sum_{c=1}^{C} {M_{ss}}^{(c)}$$
$$\text{where, } M_{tt} = \frac{1}{n_t n_t} 1_{\mathrm{t}} 1_{\mathrm{t}}^{\mathrm{T}}, \left(M_{tt}^{(c)}\right)_{ij} = \begin{cases} \dfrac{1}{n_t^{(c)} n_t^{(c)}} & x_i, x_j \in \mathrm{X}_t^{(c)} \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

where $1_s \in \mathbb{R}^{n_s}$ and $1_t \in \mathbb{R}^{n_s}$ are a column vector with all ones. From Eqs. (14), (15), (16), and (17), the balanced weighted factor, ω acts as a trade-off parameter and its value is in between $\{0,0.1,0.2\ldots1\}$, meaning that the distance matrix between domains can be optimized if we manually apply this balanced weighted factor.

Similarly, the subspace divergence can be minimized by shifting the adaptation matrices U and V to be close to one another. In our work, we employed the Frobenius norm, $\|.\|_F$, since it ensures information features data in both the source and target views to be preserved besides acting as a regularizer. The subspace divergence term can be formed as follows:

$$\min_{U,\mathrm{V}} \|U-V\|_F^2 = \min_{U,\mathrm{V}} \mathrm{Tr}\left((U-V)^T(U-V)\right) \tag{18}$$

Eq. (18) can be represented as Eq. (19) as follows:

$$\min_{U,\mathrm{V}} \mathrm{Tr}\left(\begin{bmatrix} U^T & V^T \end{bmatrix} \begin{bmatrix} \mathrm{I} & -I \\ -I & I \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix}\right) \tag{19}$$

where I is an identity matrix. U and V are the transformation matrices, and once we define U and V through joint objective function (section 3.4.4), we obtain the new representations for the source and target views, i.e., $Z_s = U^T X_s$ and $Z_t = V^T X_t$.

3.4.3 Manifold regularization

We followed the works in [5, 19, 39, 43] in adding the Laplacian regularization term. This is to take advantage of similar geometry resulting from the nearest neighbor graph, $\mathbb{G}$. Note that, manifold regularization is different from the local discriminant that was elaborated earlier (Eqs. (1)–(5)). Previously, local discriminant deals with structure in the source data, X_s, while manifold regularization acts as an additional regularizer and is added to the final view-invariant classifier (see section 4.4.4). Under the geodesic smoothness and matrix tricks, the manifold regularization is implemented in the new representation, $Z = [Z_s, Z_t]$, and is computed as follows:

$$\begin{aligned} \widehat{M_f}(P_s, P_T) &= \sum_{i,j=1}^{n_s+n_t} \left(\mathcal{A}^T z_i - \mathcal{A}^T z_j\right)^2 \widehat{W}_{ij} \\ &= Tr\left(\mathcal{A}^T Z(D-V) Z^T \mathcal{A}\right) \\ &= Tr\left(\mathcal{A}^T Z \mathcal{L} Z^T \mathcal{A}\right) \end{aligned} \tag{20}$$

where $\mathcal{L}$ is graph Laplacian matrix for manifold regularization, D is a diagonal matrix with each item $D_{ii} = \sum_{j=1}^{n} V_{ij}$ and $\mathcal{A} = (a_1, a_2)^T \in \mathbb{R}^{(n+m)} x1$ is the coefficients vector, Z. The $\widehat{W}_{ij}$ is a graph affinity matrix for new representation and is defined as:

$$\widehat{W}_{ij} = \begin{cases} \cos\left(z_i, z_j\right), & if\ z_i \in knn\left(z_j\right) or\ z_j \in knn\left(z_i\right) \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

3.4.4 Unified objective function and optimal classifier

In this last part, we included Eqs. (8), (9), (10), (13), and (19) into the joint objective function to obtain a new representation Z.

$$\max_{U,V} \frac{Tr\left(\begin{bmatrix} U^T & V^T \end{bmatrix} \begin{bmatrix} \beta(S_b + \gamma T_b) & 0 \\ 0 & \alpha S_t \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix}\right)}{Tr\left(\begin{bmatrix} U^T & V^T \end{bmatrix} \begin{bmatrix} R_{ss} + \lambda I + \beta(S_w + \gamma T_w) & R_{st} - \lambda I \\ R_{ts} - \lambda I & R_{tt} + (\lambda + \alpha) I \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix}\right)} \tag{22}$$

where , β, and λ are trade-off parameters to balance the importance of each element for each objective function, respectively. By rewriting $[U^T \quad V^T] = J^T$, and through the optimization process, we set $\delta L/\delta J = 0$, and the following equation is derived:

$$\begin{bmatrix} \beta(S_b + \gamma T_b) & 0 \\ 0 & \alpha S_t \end{bmatrix} J = \begin{bmatrix} R_{ss} + \lambda I + \beta(S_w + \gamma T_w) & R_{st} - \lambda I \\ R_{ts} - \lambda I & R_{tt} + (\lambda + \alpha) I \end{bmatrix} J\Phi \quad (23)$$

We need to solve for the eigenvector J, and once J is computed, it is easy to define the transformation matrices, U and V, and thus the new representation $Z = [Z_s, Z_t]$. Next, for the classifier, instead of using 1-NN as a standard classifier such as in [24, 38, 47], we followed [39] to learn our adaptive classifier, f, on labeled source view, $\mathcal{D}_s$ and then to predict labels in the target view, $\mathcal{D}_t$. To learn f, we summarized the structural risk minimization over D_s represented as follows:

$$f = arg \min_{f \in \mathcal{H}_K} \sum_{i=1}^{n} (y_i - f(Z_i))^2 + \eta \|f\|_K^2 \quad (24)$$

Using the representer theorem in [2], f admits the expansion:

$$f(Z) = \sum_{i=1}^{n+m} a_i K(Z_i, Z) \quad (25)$$

where $K(Z,\cdot)$ is the kernel function, a_i as in Eq. (20). We reformulated Eqs. (24) and (25) and also adding manifold regularization in Eq. (20), and as a result, the new objective function of f is defined as follows:

$$f = arg \min_{f \in \mathcal{H}_K} \|(Y - \mathcal{A}K)\theta\|_F^2 + \eta Tr\left(\mathcal{A}^T K \mathcal{A}\right) + \zeta Tr\left(\mathcal{A}^T K \mathcal{L} K^T \mathcal{A}\right) \quad (26)$$

where η and ζ are the regularized parameter, θ is the diagonal domain indicator matrix with each element $\theta_{ii} = 1$, if $i \in \mathcal{D}_s$,otherwise $\theta_{ii} = 0$. We set the derivative $\delta f / \delta \mathcal{A} = 0$, and obtain the solution of $\mathcal{A}$ as follows:

$$\mathcal{A} = ((\theta + \zeta \mathcal{L})K + \eta I)^{-1} \theta Y^T \quad (27)$$

Once we obtain $\mathcal{A}$, we can calculate $f = \mathcal{A}^* K$ (Eq. (25)) and predict the class label for $\mathcal{D}_t$. We use classification accuracy on the test data as the evaluation metric.

$$\text{Accuracy} = \frac{\left| z_t : z_t \in \mathcal{D}_t \wedge \widehat{y}(z_t) = y(x_t) \right|}{|z_t : z_t \in \mathcal{D}_t|} \quad (28)$$

where $\mathcal{D}_t$ is the target view for the new representation of target data, Z_t, $y(x_t)$ is the actual label of the target view and $\widehat{y}(z_t)$ is predicted label by the adaptive classifier, f. The complete procedural steps for our BW-UDDA model are summarized in Algorithm 1.

Algorithm 1: Balanced Weight Unified Discriminant and Distribution Alignment (BW-UDDA)

Input: Low-level feature data and source labels: X_s, X_t, Y_s ;
Parameters: $\alpha, \beta, \gamma, \lambda, \eta, \zeta$ and neighbor-p.
Adaptive Factor: $\omega = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$.

Output: Classifier, f.

Construct T_w, T_b, S_w, S_b, S_t , R_{ss}, R_{st}, R_{ts}, R_{tt}, $\mathcal{L}$ according to Eq. (4), (5), (6), (7), (10), (13), (19);
Initiate pseudo label, $\hat{y}_t$ using a classifier training on D_s.
repeat
  Solve the generalized eigen decomposition problem in Eq. (23) and obtain adaptation matrices, U and V;
  Map the original data to respective, U and V to get the new representations: $Z_s = U^T X_s$ and $Z_t = U^T X_t$;
  Compute $\mathcal{A}$ using Eq. (27) and get representer theorem in Eq. (25);
  Update the soft labels of $\mathcal{D}_t$: $\hat{y}_t = f(Z_t)$.
until *Convergence*;
Obtain the final adaptive classifier, f.

4 Experiments

In this section, we evaluated our proposed technique with two different datasets. First, in solving the open-view HAR problem, we evaluated our proposed approach with the MCAD dataset. This dataset has less correlation between views and is designed to simulate closely with the real-world surveillance scenarios. Then, to benchmark our work with a more commonly used dataset, we applied our method to the IXMAS dataset. This dataset is categorized as a constrained dataset. Although it is not characterized as an open-view dataset, this dataset has been used as a baseline for new multi-cameras action recognition methods. From these two datasets, we observed that the MCAD dataset is more challenging than that of the IXMAS dataset, and we also observed that our proposed method successfully responded to these two different datasets.

The evaluation protocol was performed in a standard cross-view validation. To the best of our knowledge, there are two methods for cross-view validation. First, the classifier is trained on one view (source view) and then tested on another view (target view) [14, 21, 51]. Second, the target view samples are split into two. The first half will be trained along with the source view, and the second half is used for testing. The second approached has been adopted in [18, 22, 27, 33]. To verify the efficacy of our method, we considered both evaluation methods and refer to them as the 1st and the 2nd cross-view evaluation method. As with other works, we adopted the leave-one-action-class-out training strategy which meant that only one action class was used for testing the target view. We noted that the classification accuracy reported here is based on the average of all action class accuracies. All conducted experiments were performed on Intel (R) CoreTM i7 system with 20GB RAM using MATLAB programming language.

4.1 MCAD dataset for open-view human action recognition

The MCAD dataset has 14,298 action samples recorded by two kinds of cameras: 3 Static cameras and 2 Pan-Tilt-Zoom (PTZ) cameras. All of the static cameras have a resolution of 1280 × 960 pixels with the fisheye effect while the PTZ cameras have a resolution of 704 × 576 pixels with a smaller field of view. The MCAD dataset complies with the open-view

action recognition criteria such as the illumination, background, day and night, field-of-view, and different split time recorded action recorded for each view [33]. The MCAD dataset has 18 action classes in total. These actions are 'point', 'wave', 'jump', 'crouch', 'sneeze', 'sit-down', 'stand-up', 'walk', 'person-run', 'cell-to-ear', 'use-cell-phone', 'drinking-water', 'take-picture', 'object-get', 'object-put', 'object-left', 'object-carry', and 'object-throw'. Figure 2 shows some example frames from the MCAD dataset.

We fixed the dimension, $d = 100$ for our method as well as for all the state-of-the-art methods. Others parameters needed for BW-UDDA were = 0.01, $\lambda = 1.0$, =0.9, =0.9, η=0.2, ζ=0.9, and maximum iteration numbers, $T = 10$. The dimensionality of the MCAD dataset was high, so we, therefore, utilized additional PCA in the pre-processing stage. All the experiments were implemented using the RBF kernel.

4.1.1 1st cross-view validation experiment

In this experiment, we evaluated our BW-UDDA and other several state-of-the-art methods using an unsupervised domain adaptation approach. We implemented a 1-NN classifier as the baseline. The state-of-the-art methods used were JDA [24], JGSA [47], MEDA [39], and JPDA [48]. We reported our recognition results in Table 2. The following are the observations we derived:

(1) Generally, results are considered poor for all methods with our proposed method achieved the highest average accuracy at 13.38%. The poor performance was due to the fact that there were no overlapping views between the training and testing data. As stated previously, the MCAD data is designed to have little correlation between views. The highest accuracy achieved for BW-UDDA technique is for PTZ06 vs. PTZ04 with

Fig. 2 Samples of the MCAD dataset from five different cameras. Each scene is different with respect to the actors, backgrounds, and views, and they are recorded in different resolutions, times, both during the day and night

an average accuracy of 19.20% while the lowest is for PTZ06 vs. Cam06 with an average of 9.29%. in the former, both cameras are of the same type with PTZ06 has a small field of view (see comment three under 2nd cross-view validation experiment). Hence, the effect of data shift is minimal. In contrast to the latter, Cam06 has a wide and distant field of view. This contrasting views between the two cameras causing the data shift to be worsen.

(2) Applying the unsupervised domain adaptation methods increased the accuracies in general. The result shows that the average accuracy for feature data without unsupervised domain adaptation is 11.32%. On the contrary, all of the unsupervised domain adaptation methods have an average accuracy higher than 11.32%.

(3) BW-UDDA outperformed the other methods in10 out of 20 experiments. The overall average for BW-UDDA, 13.38%, is higher than other state-of-the-art methods (JGSA-12.93% and JDA-12.73%). This can be explained by JGSA improving JDA in considering split adaptation matrix calculation for the dataset with very large differences for each view. While our work improves JGSA by considering the balanced weighted and local discrimination,

(4) These results confirmed that open-view action recognition is a very challenging problem and has tremendous room for improvement. With an average result of around 12% for all the evaluations, it signifies that the current research is still far from solving the real-world situation.

4.1.2 2nd cross-view validation experiment

In this experiment, we tried to observe the influence of the second half of the target view during the classification. We compared our result with SA [8], JDA [24], JGSA [47], and JPDA [48]. Results are reported in Table 3 and the followings are the observations:

(1) The overall accuracies are much better than those of the 1st experiment. This should not be a surprise since half of the target data with labels was used for training the classifier. The average accuracy for all evaluations involving all views is around 60%.

(2) The proposed technique achieved the highest performance with an average accuracy of 61.45%. It outperformed 11 out of 20 cross-view evaluations. Moreover, we also observed that our technique managed to close the accuracy variance between views while increasing the accuracy for each view. For instance, in the 1-NN baseline, the accuracy for Cam04 vs PTZ04 is 57.92%, and the accuracy for Cam04 vs PTZ06 is 63.61%. The difference between these two evaluation pairs is 5.69%. With the BW-UDDA technique, not only that the accuracy for these two pairs increases to 64.86% and 64.58%, respectively, the accuracy difference between these two evaluations decreases to only 0.28%.

(3) The PTZ06 camera always got the highest result when it acted as the target view for all methods. The reason being is that PTZ06 is physically closest to the actor compared to other cameras and thus, it has a relatively smaller field of view. This shows that different fields of view affect the accuracy performance, making MCAD a challenging dataset.

Table 2 Results for the MCAD dataset using 1st cross-view validation (train in source view and test in target view). C and P represent camera types which is 'Camera' and 'PTZ'

Src|Tgt	C4|C5	C4|C6	C4|P4	C4|P6	C5|C4	C5|C6	C5|P4	C5|P6	C6|C4	C6|C5	C6|P4	C6|P6	P4|C4	P4|C5	P4|C6	P4|P6	P6|C4	P6|C5	P6|C6	P6|P4	Ave.
1-NN	12.15	9.72	10.63	12.15	15.14	11.39	13.33	10.76	7.71	10.21	11.11	9.72	7.78	14.10	10.14	13.82	13.26	11.74	6.11	15.49	11.32
SA	13.97	11.81	9.55	17.69	14.23	11.88	**19.95**	9.79	9.11	10.19	15.43	8.20	9.80	20.34	13.33	13.52	**16.81**	11.36	7.65	16.81	13.07
JDA	15.28	10.63	10.00	**18.47**	19.72	13.75	17.71	9.79	8.33	8.96	10.56	10.83	10.07	17.78	10.21	11.67	16.53	8.82	6.04	18.47	12.68
JGSA	15.33	**12.08**	6.81	16.93	19.02	13.00	19.90	10.66	10.76	7.73	12.30	7.87	10.73	**22.97**	8.97	11.25	16.03	8.56	4.48	***23.20***	12.93
MEDA	14.89	11.27	8.72	14.10	**20.31**	8.91	18.06	4.90	9.47	**10.79**	11.36	8.38	9.90	20.79	**12.46**	15.60	14.23	8.46	6.11	21.30	12.50
JPDA	13.33	10.54	7.48	18.02	16.69	11.75	19.67	8.78	8.27	7.55	**12.30**	5.56	8.19	17.74	8.88	13.07	13.73	9.94	6.22	20.46	11.91
Ours	**16.13**	11.66	**10.39**	17.85	15.79	**13.90**	13.42	**12.09**	**10.81**	9.91	11.66	**11.29**	**12.42**	16.28	11.52	**17.05**	15.02	**11.96**	**9.29**	19.20	**13.38**

4.1.3 Confusion matrix

We next analyzed the performance of BW-UDDA for individual classes based on both the 1st and 2nd cross-view validation methods using the confusion matrices as illustrated in Fig. 3. Without loss of generality, we chose to display the results for Cam 04 vs Cam 05 in the confusion matrices. The average accuracy of the 1st cross-view was 16.13%, and the average accuracy of the 2nd cross-view was 61.39%. Here also we a saw similar observation that the accuracy of the 1st cross-view is much poorer than the accuracy of the 2nd cross-view experiments. Regardless, we found that action classes 'sit-down' and 'object thrown' obtained the highest scores in both experiments.

There were also confusing arm movements found in both confusion matrices. Examples of action classes involving arm movements are 'point', 'wave', 'cell-to-ear', 'use-cell-phone', 'drinking water', and 'take pictures'. Due to their similarity, these classes are easily confused and thus, difficult to classify. Examples of small action movements involving 'point', 'waves', 'cells to the ear', and 'use-cell-phone' from all five cameras are shown in Fig. 2.

4.2 IXMAS dataset for multi-camera constrained dataset

As mentioned earlier, we also applied our method to a standard and popular multi-camera human action recognition IXMAS dataset [42]. This dataset has 1650 action samples with 11 actions classes recorded by 4 side view cameras and 1 top-view camera. The actions involved are 'check-watch', 'cross-arms', 'get-up', 'kick', 'pick-up', 'punch', 'scratch-head', 'sit-down', 'turn-around', 'walk' and 'wave'. Following similar experimental setups as we did with the MCAD dataset, we evaluated the IXMAS dataset with 1st cross-view and 2nd cross-view validation experiments. Sample frames of the IXMAS dataset are illustrated in Fig. 4.

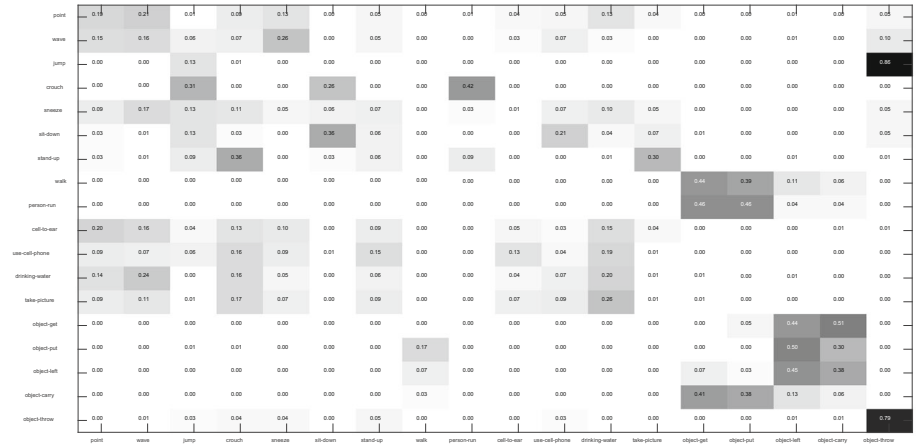
4.2.1 1st cross-view validation experiment

Once again, we used 1-NN as the baseline. We compared our results with TJM [25], TCA [28], SA [8], JDA [24], MEDA [39], and JPDA [48]. Table 4 shows all of the results and the followings are our observations:

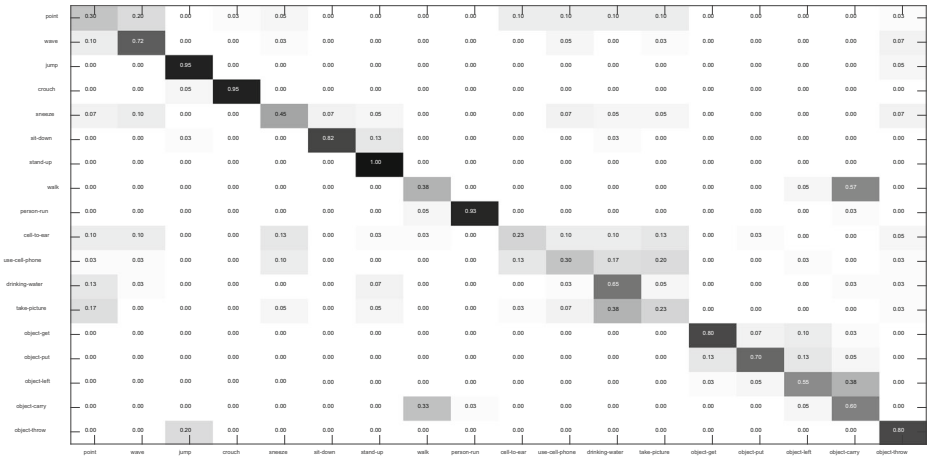
(1) As expected, the results of the 1st cross-view evaluation for IXMAS are much better compared to those of the MCAD dataset. The highest accuracy obtained is 58.18% using the JDA technique which is for Cam1 vs. Cam0. For Camera 4 (either source view or target view), the BW-UDDA method performed constantly with the highest accuracy. Unlike other cameras, camera 4 is the only camera that provides top view. The average results for all the evaluations are 27.4%. This accuracy explained that the IXMAS dataset is less challenging compared to the MCAD dataset.
(2) BW-UDDA did not perform well as expected. Our proposed method outperformed only seven out of 20 evaluations. Nonetheless, we observed that all of the seven evaluations involved Cam4 either as a source view or a target view. This indicates that our proposed technique is not affected even when the field of view is different between the cameras.

Table 3 Results for the MCAD dataset using 2nd cross-view validation (train in source view + half target view and test in another half of the target view)

Src|Tgt	C4|C5	C4|C6	C4|P4	C4|P6	C5|C4	C5|C6	C5|P4	C5|P6	C6|C4	C6|C5	C6|P4	C6|P6	P4|C4	P4|C5	P4|C6	P4|P6	P6|C4	P6|C5	P6|C6	P6|P4	Ave.
1-NN	57.78	58.06	57.92	63.61	57.22	57.78	63.33	65.14	55.14	59.58	61.39	64.03	53.89	59.31	59.44	61.94	52.92	58.89	**60.00**	62.36	59.49
SA	60.69	55.83	60.00	63.33	57.50	58.06	**64.31**	62.22	**57.22**	62.92	**64.44**	**67.08**	**56.81**	64.03	59.31	**65.83**	55.56	**61.39**	59.86	62.64	60.95
JDA	58.19	53.75	59.86	60.69	53.19	55.28	57.78	61.11	50.14	56.67	58.47	61.39	53.89	56.81	56.94	61.11	54.17	58.19	53.89	59.58	57.06
JGSA	59.17	**58.06**	58.61	63.33	57.36	55.69	63.33	63.33	55.278	58.472	61.53	61.81	55.83	61.11	57.78	63.75	55.42	60.42	56.53	60.83	59.38
JPDA	57.78	52.50	59.72	62.08	51.25	57.78	58.33	60.69	53.75	56.25	57.08	60.28	52.78	57.64	57.92	62.36	55.28	58.89	53.89	60.28	57.33
Ours	**61.39**	55.83	**64.86**	**64.58**	**57.92**	**60.14**	61.81	**67.50**	55.69	**64.17**	64.17	64.31	54.72	**65.56**	**61.67**	64.72	**56.94**	60.97	57.92	**64.17**	**61.45**



(a) Confusion matrix of the 1st cross-view evaluation experiment



(b) Confusion matrix of the 2nd cross-view evaluation experiment

Fig. 3 Analysis of BW-UDDA using confusion matrices based on the 1st cross-view evaluation and the 2nd cross-view evaluation experiments taken from Cam04 vs Cam05. Both cases are for the MCAD dataset involving 18-classes. (**a**) Confusion matrix of the 1st cross-view evaluation experiment. (**b**) Confusion matrix of the 2nd cross-view evaluation experiment

4.2.2 2nd cross-view validation experiment

Similarly, we compared our proposed technique with SDA [34], TJM [25], TCA [28], SA [8], JDA [24], JGSA [47], and JPDA [48], and all of the results are tabulated in Table 5 with the following observations:

(1) Similar to the MCAD dataset performance, 2nd cross-view validation experiment showed much better results. The overall average accuracy is around 85%. This value is higher than that of the MCAD because the IXMAS dataset is less challenging.

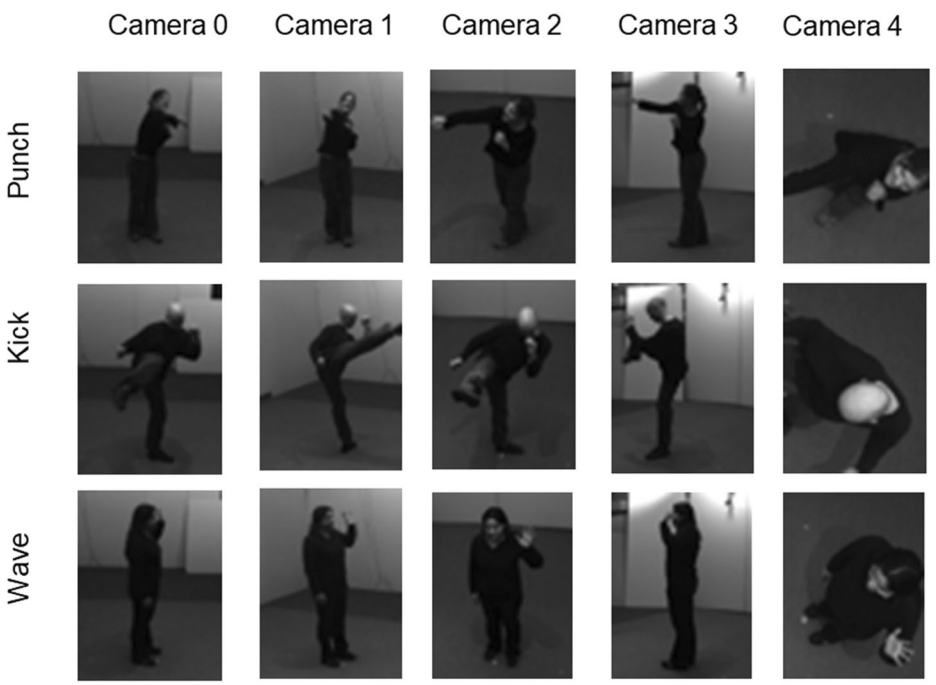


Fig. 4 Samples of IXMAS multi-view dataset. Each row shows action viewed across five different cameras

(2) BW-UDDA showed the highest performance with an average accuracy of 90.91%. As can be seen, BW-UDDA outperformed in 17 out of 20 evaluations. This result confirms the potential of the proposed method in unsupervised domain adaptation.

4.2.3 Confusion matrix

We further analyzed each class from both evaluations of the IXMAS dataset in a form of a confusion matrix. Figure 5 shows the confusion matrix for Cam4 vs Cam3. The average accuracy for the 1st cross-view evaluation was 20% while the average accuracy for the 2nd cross-view evaluation was 95.15%. From both matrices, class-action 'pick-up' and 'sit-down' were consistently well classified. However, from Fig. 5(b), 'punch' and 'wave' were easily confused with the 'kick' action. This may be explained that even in a 'kick' action the arm movement is still involved, and the confusion is exacerbated by the low-resolution frames of the IXMAS dataset.

4.3 Balanced weighted factor analysis

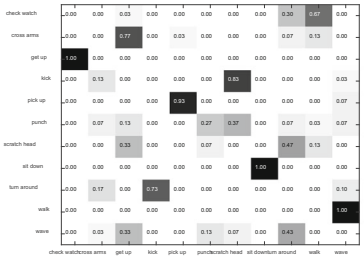
As stated in section 4.3, BW-UDDA depends on the balanced weighted factor, ω, to obtain the optimum accuracy. To show its influence on the accuracy performance we analyzed several values of ω. Specifically, we use ω in range = {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 and 1.0}. For the analysis, we selected the 2nd cross-view evaluation in MCAD dataset. We chose

Table 4 Results for the IXMAS dataset using 1st cross-view validation (train in source view and test in target view)

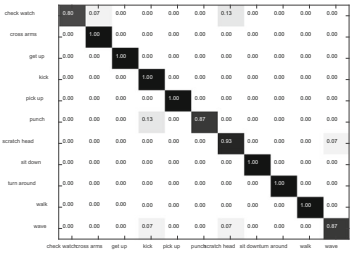
Src|Tgt	C0|C1	C0|C2	C0|C3	C0|C4	C1|C0	C1|C2	C1|C3	C1|C4	C2|C0	C2|C1	C2|C3	C2|C4	C3|C0	C3|C1	C3|C2	C3|C4	C4|C0	C4|C1	C4|C2	C4|C3	Ave.
1-NN	37.27	36.36	24.85	15.76	34.85	15.45	49.09	12.42	37.27	22.12	29.39	13.94	28.79	56.67	26.97	10.30	17.88	10.00	17.27	10.61	25.36
TJM	45.76	40.61	46.06	17.88	49.09	14.85	59.39	12.42	40.00	20.30	36.36	16.97	35.76	56.06	26.36	13.94	23.03	10.00	15.45	11.52	29.59
TCA	46.67	39.70	43.94	16.06	47.88	15.15	58.18	12.73	**40.91**	13.94	23.64	16.97	37.58	54.24	26.06	11.52	22.42	9.39	15.76	11.52	28.21
SA	40.00	36.67	29.09	14.55	37.88	**13.64**	51.82	10.91	38.79	**20.30**	29.70	16.36	31.82	54.24	25.15	13.33	18.48	9.39	15.76	11.82	25.98
JDA	**58.79**	34.85	**53.94**	16.97	**58.18**	9.39	**58.18**	10.91	40.61	12.42	23.64	16.67	**47.27**	**57.58**	**31.52**	10.00	19.70	9.70	17.58	10.91	**29.94**
JGSA	52.42	**42.12**	37.58	**21.82**	55.76	7.88	58.48	15.45	39.394	18.788	**35.758**	16.061	41.82	55.45	25.76	10.91	20.61	6.97	18.79	13.03	29.74
MEDA	41.52	35.15	26.06	19.70	43.33	7.58	50.61	9.70	37.88	7.58	29.39	17.27	27.27	53.33	27.88	11.82	16.36	9.70	17.58	11.82	25.08
JPDA	32.73	38.48	31.82	16.06	34.55	9.70	52.12	17.88	40.91	11.82	17.88	17.27	29.70	53.64	15.76	10.00	22.73	9.39	18.48	12.73	24.68
Ours	43.03	39.39	28.18	19.09	50.91	7.27	52.12	**18.18**	36.06	16.06	32.73	**17.58**	28.79	53.03	27.27	**20.91**	**23.94**	**10.30**	**23.33**	**20.00**	28.41

Table 5 Results for the IXMAS dataset using 2nd cross-view validation (train in source view + half target view and test in another half target view)

Src|Tgt	C0|C1	C0|C2	C0|C3	C0|C4	C1|C0	C1|C2	C1|C3	C1|C4	C2|C0	C2|C1	C2|C3	C2|C4	C3|C0	C3|C1	C3|C2	C3|C4	C4|C0	C4|C1	C4|C2	C4|C3	Ave.
1-NN	75.76	73.94	80.00	84.85	81.21	73.33	86.06	78.18	83.03	76.97	81.82	87.27	83.64	76.97	80.00	81.82	82.42	73.94	74.55	86.06	80.09
SDA	80.61	79.39	86.06	83.64	84.24	81.21	87.88	86.67	88.48	83.03	90.91	88.48	89.70	76.36	77.58	85.45	81.82	78.79	76.36	88.48	83.76
TJM	84.85	86.06	90.91	90.91	88.48	84.85	90.30	90.30	84.85	80.61	88.48	86.06	87.88	84.85	87.88	87.27	82.42	83.64	85.45	84.85	86.55
TCA	80.61	80.61	89.09	87.27	85.45	83.64	86.06	81.21	87.88	81.21	83.64	85.45	83.64	86.67	85.45	87.27	88.48	82.42	85.45	89.09	85.03
SA	83.03	79.39	90.30	87.88	87.27	78.79	86.67	85.45	87.27	79.39	89.70	92.12	87.27	80.00	79.39	85.45	83.64	84.24	76.36	86.67	84.52
JDA	86.67	80.00	87.88	84.24	84.85	81.21	91.52	88.48	84.24	75.76	84.85	87.27	84.85	81.21	84.24	87.88	86.67	84.24	83.03	89.09	84.91
JGSA	86.67	82.42	**92.73**	90.30	89.09	84.85	92.73	90.30	86.061	85.455	89.697	87.879	**91.52**	84.85	79.39	86.67	89.70	**86.67**	81.21	93.33	87.58
JPDA	81.21	82.42	90.91	86.06	86.67	80.61	85.45	92.12	84.24	82.42	89.09	83.64	87.27	83.64	84.24	83.64	**90.91**	82.42	82.42	87.88	85.36
Ours	**89.70**	**86.67**	91.52	**92.73**	**94.55**	**86.06**	**95.76**	**96.97**	**92.73**	**87.27**	**93.33**	**96.36**	87.27	**90.30**	**84.85**	**94.55**	**90.91**	86.06	**85.45**	**95.15**	**90.91**



(a) Confusion matrix of the 1st cross-view evaluation experiment



(b) Confusion matrix of the 2nd cross-view evaluation experiment

Fig. 5 Analysis of BW-UDDA using confusion matrices based on the 1st cross-view evaluation and the 2nd cross-view evaluation experiments taken from Cam04 vs Cam03. Both cases used the IXMAS dataset involving 11-classes. (**a**) Confusion matrix of the 1st cross-view evaluation experiment. (**b**) Confusion matrix of the 2nd cross-view evaluation experiment

Cam04 as the source view and the rest as target views. The results are plotted and shown in Fig. 6.

Traditionally, it has been assumed that the marginal and conditional distribution are equally important. This scenario is similar to setting the balanced weighted factor, $\omega = 0.5$. However, as can be seen in Fig. 6, this value of ω did not perform satisfactorily. On one extreme, if the value of ω is set to 0, the overall performance drops dramatically. This indicates that both the conditional and marginal distributions cannot be ignored in the BW-UDDA settings. On the other extreme, when the value of ω is set to 1, most of the cross-view evaluations reached their

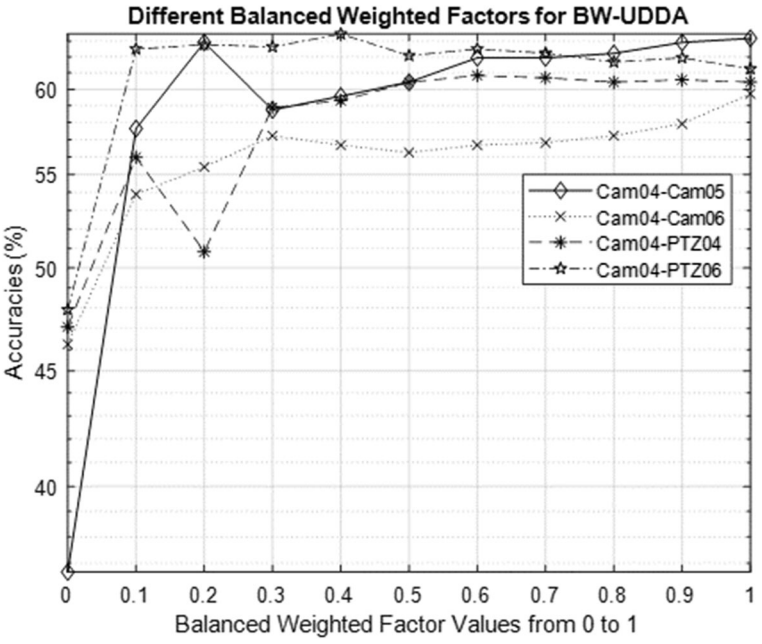


Fig. 6 Balanced weighted factor ω with different values, and the optimal accuracy for BW-UDDA using the MCAD dataset

Table 6 Accuracy analysis (%) for BW-UDDA with and without LSDA via MCAD dataset

Src|Tgt	C4|C5	C4|C6	C4|P4	C4|P6	C5|C4	C5|C6	C5|P4	C5|P6	C6|C4	C6|C5	C6|P4	C6|P6	P4|C4	P4|C5	P4|C6	P4|P6	P6|C4	P6|C5	P6|C6	P6|P4	Ave.
With LSDA	**61.1**1	56.25	**63.61**	**65.42**	**57.22**	**59.03**	60.56	**64.31**	**55.00**	**61.39**	**56.81**	**62.64**	**55.83**	**65.00**	**59.86**	63.75	**57.78**	57.78	**56.25**	61.94	**61.11**
Without LSDA	57.78	**59.31**	62.22	63.06	56.39	55.00	**64.03**	61.25	48.75	60.42	56.53	62.08	49.44	59.03	56.94	**64.03**	52.22	**59.44**	53.89	**62.08**	57.78

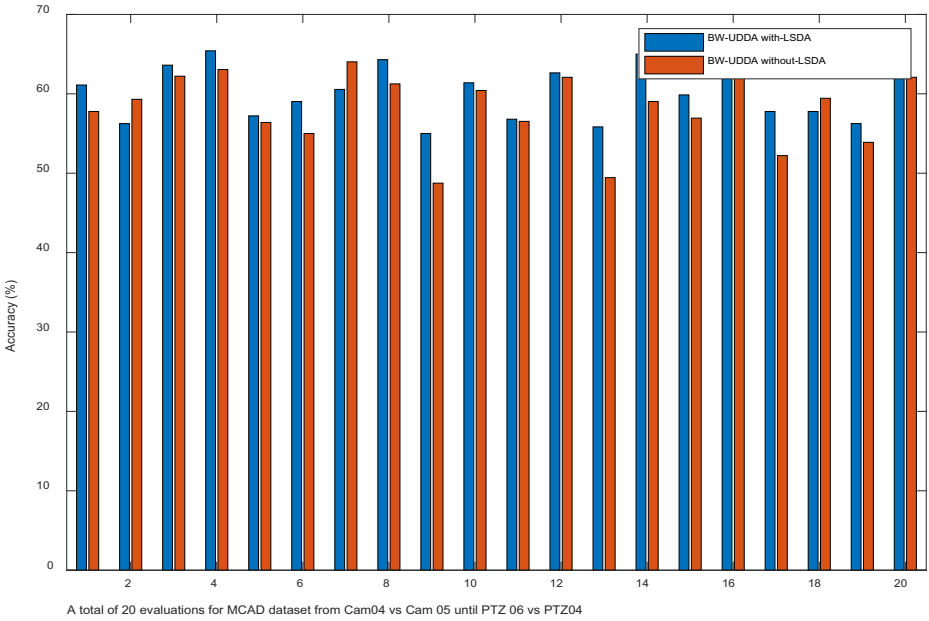


Fig. 7 Visualization comparison of BW-UDDA between with and without LSDA

optimum accuracy. This is a situation in which the conditional distribution contributes more to the overall performance than that of the marginal distribution. However, this scenario is not uniform throughout all evaluations. As an example, in Cam 04 vs PTZ 06 evaluation, the accuracy performance is optimal when the value of ω is set to 0.4, and the accuracy decreases as we increase the value of ω to 1.

4.4 Local discriminant effectiveness analysis

Finally, we analyzed the effectiveness and advantages of the LSDA. First, we assessed the BW-UDDA with and without LSDA. To perform the analysis, we fixed the balanced weighted factor, ω, to 0.5. Again, we selected the 2nd cross-view evaluation using the MCAD dataset for the analysis. We recorded the results in Table 6 and plotted the comparison results in Fig. 7. From Table 6 and Fig. 7, we observed that BW-UDDA with LSDA showed a significant influence in the accuracy compared to BW-UDDA without LSDA. Out of 20, BW-UDDA outperformed in 15 evaluations and has the highest average accuracy of 60.08%. These results established the importance of incorporating LSDA into our proposed technique and confirmed the work of [23] that combining LDA with LSDA improves accuracy.

5 Conclusion

This paper deals with how to improve the human action recognition (HAR) field that suffers from data shifts problem due to the large differences between data distributions of the target and source views. Such a problem degrades significantly the performance accuracy particularly in an unconstrained dataset for open view HAR case. To alleviate this problem, we

leveraged the unsupervised domain adaptation method to reduce the data shift problem and to increase the accuracy performance. Specifically, we proposed Balanced Weighted-Unified Discriminant and Distribution Alignment (BW-UDDA) to improve the unsupervised domain adaptation technique for open-view HAR. The outcomes of experiments we conducted proved that our proposed method outperformed most of the state-of-the-art unsupervised domain adaptation methods when applied to open-view HAR. The results also indicated that the open-view HAR remains to be a challenging problem. Therefore, ongoing efforts to further improve and enhance the performance will continue as HAR places more useful applications in our daily lives.

Funding The authors thank the Ministry of Education Malaysia and University Technology Malaysia (UTM) for their support under the Fundamental Research Scheme, grant number R.J130000.7851.5F179.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Aggarwal JK, Ryoo MS (2011) Human activity analysis: a review. ACM Comput Surv 43:1–43. https://doi.org/10.1145/1922649.1922653
2. Belkin M, Niyogi P, Sindhwani V (2006) Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. J Mach Learn Res 7(11)
3. Li B, Camps OI, Sznaier M (2012) Cross-view activity recognition using Hankelets. IEEE Conference on Computer Vision and Pattern Recognition. IEEE, In, pp 1362–1369. https://doi.org/10.1109/CVPR.2012.6247822
4. Gong B, Shi Y, Sha F, Grauman K (2012) Geodesic flow kernel for unsupervised domain adaptation. In: 2012 IEEE conference on computer vision and pattern recognition. IEEE 2066–2073. doi: https://doi.org/10.1109/CVPR.2012.6247911
5. Cai J, Huang X (2018) Modified sparse linear-discriminant analysis via nonconvex penalties. IEEE Trans Neural Networks Learn Syst 29:4957–4966. https://doi.org/10.1109/TNNLS.2017.2785324
6. Ciptadi A, Goodwin MS, Rehg JM (2014) Movement pattern histogram for action recognition and retrieval. Eur Conf Comput Vision:695–710. https://doi.org/10.1007/978-3-319-10605-2_45
7. Farhadi A, Tabrizi MK (2008) Learning to recognize activities from the wrong view point. In: European conference on computer vision. Springer, Berlin, Heidelberg. 154–166. https://doi.org/10.1007/978-3-540-88682-2_13
8. Fernando B, Habrard A, Sebban M, Tuytelaars T (2013) Unsupervised visual domain adaptation using subspace alignment. In: 2013 IEEE international conference on computer vision. IEEE, pp 2960–2967. https://doi.org/10.1109/ICCV.2013.368
9. Ghifary M, Balduzzi D, Kleijn WB, Zhang M (2017) Scatter component analysis: a unified framework for domain adaptation and domain generalization. IEEE Trans Pattern Anal Mach Intell 39:1414–1430. https://doi.org/10.1109/TPAMI.2016.2599532
10. Gorelick L, Blank M, Shechtman E, Member S, Irani M, Basri R (2007) Action as space time shapes. IEEE Trans Pattern Anal Mach Intell 29:2247–2253. https://doi.org/10.1109/TPAMI.2007.70711
11. Junejo IN, Dexter E, Laptev I, Pérez P (2011) View-independent action recognition from temporal self-similarities. IEEE Trans Pattern Anal Mach Intell 33:172–185. https://doi.org/10.1109/TPAMI.2010.68
12. Junejo IN, Dexter E, Laptev I, Púrez P (2008) Cross-view action recognition from temporal self-similarities. In: European conference on computer vision. Springer, Berlin, Heidelberg, pp. 293–306. https://doi.org/10.1109/TPAMI.2010.68, 33
13. Kase N, Babaee M, Rigoll G (2017) Multi-view human activity recognition using motion frequency. In: IEEE international conference on image processing (ICIP). IEEE, pp 3963–3967. https://doi.org/10.1109/TIP.2017.2696786
14. Kong Y, Ding Z, Li J, Fu Y (2017) Deeply learned view-invariant features for cross-view action recognition. IEEE Trans Image Process 26:3028–3037. https://doi.org/10.1109/TIP.2017.2696786

15. Kulathumani V, Kavi R, Ramagiri S (2011) WVU multi-view action recognition dataset. Available on: http://csee.WVUEdu/~vkkulathumani/WVU-action.Html# download2.
16. Laptev L (2003) Space-time interest points. IEEE International Conference on Computer Vision. IEEE, In, pp 432–439. https://doi.org/10.1109/ICCV.2003.1238378
17. Li R, Zickler T (2012) Discriminative virtual views for cross-view action recognition. In: IEEE computer society conference on computer vision and pattern recognition. 2855–2862. Pp 187–196. https://doi.org/10.1109/WACV.2017.28
18. Li W, Wong Y, Liu AA, Li Y, Su YT, Kankanhalli M (2017) Multi-camera action dataset for cross-camera action recognition benchmarking. IEEE Winter Conf Appl Comput Vision, WACV 2017:187–196. https://doi.org/10.1109/ICME.2019.00124
19. Li Y, Cheng L, Peng Y, Wen Z, Ying S (2019) Manifold alignment and distribution adaptation for unsupervised domain adaptation. IEEE International Conference on Multimedia and Expo, In, pp 688–693. https://doi.org/10.1109/CVPR.2011.5995729
20. Liu J, Shah M, Kuipers B, Savarese S (2011) Cross-view action recognition via view knowledge transfer. IEEE Conference on Computer Vision and Pattern Recognition. IEEE, In, pp 3209–3216. https://doi.org/10.1109/TCSVT.2018.2868123
21. Liu Y, Lu Z, Li J, Yang T (2019) Hierarchically learned view-invariant representations for cross-view action recognition. IEEE Transn Circ Syst Video Technol 29:2416–2430. https://doi.org/10.1109/TCSVT.2018.2868123
22. Liu Y, Lu Z, Li J, Yao C, Deng Y (2018) Transferable feature representation for visible-to-infrared cross-dataset human action recognition. Complexity 2018:1–20. https://doi.org/10.1155/2018/5345241
23. Liu Z, Liu G, Pu J, Wang X, Wang H (2018) Orthogonal sparse linear discriminant analysis. Int J Syst Sci 49:847–857. https://doi.org/10.1080/00207721.2018.1424964
24. Long M, Wang J, Ding G, Sun J, Yu PS (2013) Transfer feature learning with joint distribution adaptation. IEEE International Conference on Computer Vision, In, pp 2200–2207. https://doi.org/10.1109/CVPR.2014.183
25. Long M, Wang J, Ding G, Sun J, Yu PS (2014) Transfer joint matching for unsupervised domain adaptation. IEEE Conference on Computer Vision and Pattern Recognition. IEEE, In, pp 1410–1417. https://doi.org/10.1049/iet-cvi.2015.0416
26. Murtaza F, Yousaf MH, Velastin SA (2016) Multi-view human action recognition using 2D motion templates based on MHIs and their HOG description. IET Comput Vis 10:758–767. https://doi.org/10.1049/iet-cvi.2015.0416
27. Nie W, Liu A, Yu J, Su Y, Chaisorn L, Wang Y, Kankanhalli MS (2014) Multi-view action recognition by cross-domain learning. In: international workshop on multimedia signal processing (MMSP). IEEE, pp 1–6. https://doi.org/10.1109/TNN.2010.2091281
28. Pan SJ, Tsang IW, Kwok JT, Yang Q (2011) Domain adaptation via transfer component analysis. IEEE Trans Neural Netw 22:199–210. https://doi.org/10.1109/TNN.2010.2091281
29. Pan SJ, Yang Q (2010) A survey on transfer learning. IEEE Trans Knowl Data Eng 22:1345–1359. https://doi.org/10.1007/978-981-15-5971-6_83
30. Peng X, Zou C, Qiao Y, Peng Q (2014) Action recognition with stacked fisher vectors. In: lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics). Pp 581–595. https://doi.org/10.1109/ICIP.2017.8297026
31. Shao L, Member S, Zhu F, Member S, Li X (2015) Transfer learning for visual categorization : a survey. IEEE Trans Neural Networks Learn Syst 26:1019–1034. https://doi.org/10.1109/TNNLS.2014.2330900
32. Singh S, Velastin SA, Ragheb H (2010) MuHAVi: a multicamera human action video dataset for the evaluation of action recognition methods. IEEE International Conference on Advanced Video and Signal Based Surveillance. IEEE, In, pp 48–55. https://doi.org/10.1109/AVSS.2010.63
33. Su Y, Li Y, Liu A (2019) Open-view human action recognition based on linear discriminant analysis. Multimed Tools Appl 78:767–782. https://doi.org/10.1007/s11042-018-5657-6
34. Sun B, Saenko K (2015) Subspace distribution alignment for unsupervised domain adaptation. In: Procedings of the British machine vision conference 2015. British Mach Vision Assoc 24:1–24.10. https://doi.org/10.5244/c.29.24
35. Wang H, Kläser A, Schmid C, Liu CL (2011) Action recognition by dense trajectories. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, In, pp 3169–3176. https://doi.org/10.1109/CVPR.2011.5995407
36. Wang H, Schmid C (2013) Action recognition with improved trajectories. IEEE International Conference on Computer Vision. IEEE, In, pp 3551–3558. https://doi.org/10.1109/ICCV.2013.441
37. Wang J, Chen Y, Feng W, Han YU, Huang M, Yang Q (2020) Transfer learning with dynamic distribution adaptation. ACM transactions on intelligent systems and technology (TIST), pp 1–25. https://doi.org/10.1145/3360309

38. Wang J, Chen Y, Hao S, Feng W, Shen Z (2017) Balanced distribution adaptation for transfer learning. In: IEEE international conference on data mining (ICDM). IEEE, pp 1129–1134. https://doi.org/10.1109/ICDM.2017.150
39. Wang J, Feng W, Chen Y, Yu H, Huang M, Yu PS (2018) Visual domain adaptation with manifold embedded distribution alignment. In: proceedings of the 26th ACM international conference on multimedia. Pp 402–410. https://doi.org/10.1145/3240508.3240512
40. Wang J, Yang J, Yu K, Lv F, Huang T, Gong Y (2010) Locality-constrained linear coding for image classification. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, In, pp 3360–3367. https://doi.org/10.1109/CVPR.2010.5540018
41. Weinland D, Boyer E, Ronfard R (2007) Action recognition from arbitrary views using 3D exemplars. In: IEEE 11th international conference on computer vision. IEEE, pp 1–7.
42. Weinland D, Weinland D, Weinland D, Ronfard R (2006) Free viewpoint action recognition using motion history volumes. Comput Vis Image Underst 104(2–3):249–257
43. Wen J, Fang X, Cui J, Fei L, Yan K, Chen Y, Xu Y (2019) Robust sparse linear discriminant analysis. IEEE Trans Circ Syst Video Technol 29:390–403. https://doi.org/10.1109/TCSVT.2018.2799214
44. Wu X, Wang H, Liu C, Jia Y (2015) Cross-view action recognition over heterogeneous feature spaces. IEEE Trans Image Process 24:4096–4108. https://doi.org/10.1109/TIP.2015.2445293
45. Yan Y, Ricci E, Subramanian R, Liu G, Sebe N (2014) Multitask linear discriminant analysis for view invariant action recognition. IEEE Trans Image Process 23:5599–5611. https://doi.org/10.1109/TIP.2014.2365699
46. Yang Y, Hospedales T (2015) Zero-shot domain adaptation via kernel regression on the Grassmannian. In: Proceedings the 1st international workshop on differential geometry in computer vision for analysis of shapes. BMVA Press, Images and Trajectories, pp 1.1–1.12
47. Zhang J, Li W, Ogunbona P (2017) Joint geometrical and statistical alignment for visual domain adaptation. IEEE Conference on Computer Vision and Pattern Recognition. IEEE, In, pp 5150–5158
48. Zhang W, Wu D (2020) Discriminative joint probability maximum mean discrepancy (DJP-MMD) for domain adaptation. Proceedings of the International Joint Conference on Neural Networks, In, pp 1–8. https://doi.org/10.1109/CVPR.2017.547
49. Zhang Z, Wang C, Xiao B, Zhou W, Liu S (2014) Cross-view action recognition using contextual maximum margin clustering. IEEE Trans Circ Syst Video Technol 24:1663–1668. https://doi.org/10.1109/TCSVT.2014.2305552
50. Zhang Z, Wang C, Xiao B, Zhou W, Liu S, Shi C (2013) Cross-view action recognition via a continuous virtual path. In proceedings of the IEEE conference on computer vision and pattern recognition 2690–2697. https://doi.org/10.1109/CVPR.2013.347
51. Zheng J, Jiang Z, Chellappa R (2016) Cross-view action recognition via transferable dictionary learning. IEEE Trans Image Process 25:2542–2556. https://doi.org/10.1109/TIP.2016.2548242
52. Zheng J, Jiang Z, Phillips J, Chellappa R (2012) Cross-view action recognition via a transferable dictionary pair. British Machine Vision Conference, In, pp 125.1–125.11
53. Zhu F, Shao L (2014) Weakly-supervised cross-domain dictionary learning for visual recognition. Int J Comput Vis 109:42–59. https://doi.org/10.1007/s11263-014-0703-y