Domain Adaptive Fisher Vector for Visual Recognition

Li Niu[1](✉), Jianfei Cai[2], and Dong Xu[3]

[1] Interdisciplinary Graduate School,
Nanyang Technological University, Singapore, Singapore
lniu002@ntu.edu.sg
[2] School of Computer Engineering,
Nanyang Technological University, Singapore, Singapore
asjfcai@ntu.edu.sg
[3] School of Electrical and Information Engineering,
University of Sydney, Sydney, Australia
dong.xu@sydney.edu.au

Abstract. In this paper, we consider Fisher vector in the context of domain adaptation, which has rarely been discussed by the existing domain adaptation methods. Particularly, in many real scenarios, the distributions of Fisher vectors of the training samples (*i.e.*, source domain) and test samples (*i.e.*, target domain) are considerably different, which may degrade the classification performance on the target domain by using the classifiers/regressors learnt based on the training samples from the source domain. To address the domain shift issue, we propose a Domain Adaptive Fisher Vector (DAFV) method, which learns a transformation matrix to select the domain invariant components of Fisher vectors and simultaneously solves a regression problem for visual recognition tasks based on the transformed features. Specifically, we employ a group lasso based regularizer on the transformation matrix to select the components of Fisher vectors, and use a regularizer based on the Maximum Mean Discrepancy (MMD) criterion to reduce the data distribution mismatch of transformed features between the source domain and the target domain. Comprehensive experiments demonstrate the effectiveness of our DAFV method on two benchmark datasets.

Keywords: Domain adaptation · Fisher vector

1 Introduction

Constructing global feature representations based on local descriptors of images/videos is a common approach in a multitude of visual recognition tasks. As a commonly used encoding method, Fisher vector [1] encodes both first and second order statistical information of local descriptors *w.r.t.* the generative model (*e.g.*, Gaussian Mixture Model (GMM)) trained based on them, and one Gaussian model in the GMM corresponds to one component in the extracted

B. Leibe et al. (Eds.): ECCV 2016, Part VI, LNCS 9910, pp. 550–566, 2016.
DOI: 10.1007/978-3-319-46466-4_33

Fisher vector. Recently, Fisher vector achieves excellent performance for object recognition [2–5] or human action recognition [6,7]. To extract Fisher vector, we generally train a GMM based on the local descriptors of training samples and extract Fisher vectors for both training and test samples based on the pre-trained GMM. However, the GMM trained on the training samples does not consider the data distribution of test samples properly and thus lacks the generalization ability [8] on the test samples, leading to unsatisfactory recognition performance on the test datasets.

According to the terminology in the field of domain adaptation, the training dataset and the test dataset are referred to as the source domain and the target domain, respectively. When the target domain data are available in the training stage, we can train GMMs based on the mixture of local descriptors from both source domain and target domain. However, even in this case, the generated Fisher vectors of source domain samples and target domain samples may be still considerably different in terms of statistical properties, which is referred to as dataset bias [9]. Instead of training GMMs based on the data from both domains, another approach is to adapt the GMM trained based on the source domain to the target domain [8], or interpolate two GMMs which are trained based on the source domain and the target domain separately [10]. However, these methods did not explicitly consider the domain distribution mismatch between the source domain and the target domain. So they cannot guarantee the extracted Fisher vectors based on the adapted or interpolated GMMs are domain invariant.

In recent works, many domain adaptation approaches [11–20] have been proposed to tackle the domain shift issue between the source domain and the target domain (see Sect. 2 for details). However, none of them is specifically designed for Fisher Vector, since they did not take the generative models (*i.e.*, GMMs) into consideration. Therefore, the excellent performance of Fisher vector for visual recognition [5,7] and the lack of effective domain adaptation methods for Fisher vector motivate our work. By noticing that each Gaussian model in the GMM characterizes the data distribution of a cluster of local descriptors, and some Gaussian models are more likely to capture the common data distribution between the source domain and the target domain, we come out the idea of identifying the common Gaussian models via selecting the corresponding components of Fisher vectors that are more likely to be domain invariant.

Let us take the object recognition and human action recognition tasks as two examples to provide more explanations for domain invariant components. For object recognition, the appearance of images within the same category may be quite different between the source domain and the target domain, which is usually referred to as intra-class difference, while some specific object regions within the category may be relatively consistent. Considering extracting the CNN features of object proposals as local descriptors and encoding them into Fisher vectors based on the pre-trained GMM, we expect to select the components of Fisher vectors corresponding to the Gaussian models from the object proposals which are more consistent across the source domain and the target domain. To validate this point, we present a detailed showcase associated with more discussions in Sect. 5.1. For human action recognition, sometimes the videos in the source domain are captured from the front view while the videos in the target domain are captured from the back view. When using the popular Improved Dense Trajectory (IDT) features as local descriptors in videos, each trajectory represents a local movement of human body, some of which can be observed from both front view and back view while the others can only be observed from one view. After encoding the IDT descriptors in videos into Fisher vectors based on the pre-trained GMM, we want to select the components of Fisher vectors corresponding to the Gaussian models from the trajectories which can be observed from both views.

To this end, we propose our Domain Adaptive Fisher Vector (DAFV) method. Specifically, we learn a transformation matrix to project the Fisher vectors into a lower dimensional latent subspace and consider visual recognition task as a regression problem based on the transformed features. A group lasso based regularizer [21] is employed on the transformation matrix to enforce the components of the transformation matrix corresponding to the selected (*resp.*, unselected) components of Fisher vectors to be associated with large (*resp.*, small) weights. At the same time, we apply the criterion of minimizing the Maximum Mean Discrepancy (MMD) of the transformed features between the source domain and the target domain by using an MMD-based regularizer. In Sect. 3, we briefly provide the background knowledge of Fisher vector. In Sect. 4, we introduce our Domain Adaptive Fisher Vector (DAFV) method in detail and also present a novel solution to the nontrivial optimization problem. In Sect. 5, we conduct extensive experiments on two benchmark datasets Bing-Caltech256 and ACT4^2 to demonstrate the effectiveness of our proposed method.

Our major contributions can be summarized as follows: (1) to the best of our knowledge, domain adaptation method designed for Fisher vectors has been rarely discussed in the previous literature. This is the first work to select domain invariant components of Fisher vectors to reduce the domain distribution mismatch between the source domain and the target domain; (2) we propose a Domain Adaptive Fisher Vector (DAFV) method and develop an effective solution to the proposed formulation; (3) extensive experiments on two benchmark datasets show the effectiveness of our method for selecting domain invariant components.

2 Related Work

Our work is related to using Fisher vector for visual recognition tasks. Fisher vector was first used for image classification in [22] and further improved in [2] with power normalization and L_2 normalization. In [3], Simonyan *et al.* developed a two-layer deep network based on Fisher vector for large-scale image classification. More recently, with the breakthrough in image representation by using Convolutional Neural Networks (CNN), CNN features of local regions have been used as local descriptors for Fisher vector [4,5,23,24]. Fisher vector was also applied to video action and event recognition [6,25]. Similar to the idea in [3] for image classification, Peng *et al.* proposed stacked Fisher vectors for human action recognition in [7]. All these methods assume the training samples and test samples are with the same data distribution while this assumption does not hold in domain adaptation scenarios.

Our work is related to domain adaptation. The existing domain adaptation methods can be classified into feature-based methods [13–18,26–28], SVM-based methods [12,29–32], instance-reweighting methods [11], dictionary learning methods [19], and low-rank based methods [20,33]. All the above methods are not specifically designed for Fisher vector. Among them, our method is more related to [16] and [17] which also learn a transformation matrix. However, [16,17] are only feature learning methods without considering the property of Fisher vector while our method can select the domain invariant components of Fisher vectors and simultaneously learn the regression matrix.

Finally, our work is also related to adapted or interpolated GMMs. Recently, Bayesian model adaptation has attracted much attention and several approaches have been proposed to adapt the background GMM to each image [34] or each category with very few examples [35]. Then, a more general formulation of Bayesian adaptation was proposed in [8] for image classification. Note that these methods [8,34,35] focus on adapting the background GMM to either a new image or a new category instead of considering the difference between two domains. So the motivation of their methods is intrinsically different from ours. More recently, Kim *et al.* proposed to interpolate a set of GMMs on the manifold in [10], which can be used to learn the interpolation between two GMMs from two domains. Nevertheless, all the above works did not explicitly address the domain shift issue. In contrast, our method explicitly reduces the domain distribution mismatch between two domains. Moreover, the Fisher vectors based on the GMMs learnt by their methods can be readily used to replace the original Fisher vectors in our method to further improve the performance.

3 Fisher Vector

In the remainder of this paper, we denote a matrix/vector by using a uppercase/lowercase letter in boldface (*e.g.*, $\mathbf{A}$ denotes a matrix and $\mathbf{a}$ denotes a vector). We denote an n-dim column vector of all zeros and all ones by using $\mathbf{0}_n, \mathbf{1}_n \in \mathbb{R}^n$, respectively. Note that when the dimension is obvious, we use $\mathbf{0}$ and $\mathbf{1}$ instead of $\mathbf{0}_n$ and $\mathbf{1}_n$ for simplicity. We use $\mathbf{I}$ to denote identify matrix. The superscript $'$ is used to denote the transpose of a matrix or a vector. Moreover, we use $\mathbf{A}^{-1}$ to denote the inverse matrix of $\mathbf{A}$ and $\mathbf{A} \circ \mathbf{B}$ to denote the element-wise product between two matrices $\mathbf{A}$ and $\mathbf{B}$.

Fisher vector is a commonly used encoding method to construct global feature representations from local descriptors. As a combination of generative and discriminative approaches, on one hand, the generation procedure of a set of local descriptors $\mathbf{X} = \{\mathbf{x}_i|_{i=1}^{N}\}$ (N is the number of local descriptors) is assumed to obey a probability density function $p(\mathbf{X}; \boldsymbol{\theta})$ with parameters $\boldsymbol{\theta}$. On the other hand, the gradients of the log-likelihood *w.r.t.* the model parameters, which describe the contribution of model parameters to the generation procedure of $\mathbf{X}$ [1], can be used as input features for discriminative methods such as classifiers and regressors. Since each image/video can be treated as a set of local descriptors $\{\mathbf{x}_i|_{i=1}^{N}\}$, its Fisher vector can be represented as,

$$G_{\boldsymbol{\theta}}^{\mathbf{X}} = \frac{1}{N}\sum_{i=1}^{N} \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}_i; \boldsymbol{\theta}). \tag{1}$$

For visual recognition tasks, the probability density function $p(\mathbf{X}; \boldsymbol{\theta})$ is usually modeled by Gaussian Mixture Model (GMM) [22,25]. Suppose K is the number of Gaussian models in the GMM, we use model parameters $\boldsymbol{\theta} = \{\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\sigma}_1; \ldots; \pi_K, \boldsymbol{\mu}_K, \boldsymbol{\sigma}_K\}$ to denote the mixture weights, means, and diagonal covariances of GMM, respectively. Based on the definition of Fisher vector (1), the gradients of the log-likelihood *w.r.t.* the model parameters (*i.e.*, means and diagonal covariances) of the k-th Gaussian model can be written as (refer to [22] for the derivation details),

$$\mathcal{G}_{\boldsymbol{\mu},k}^{\mathbf{X}} = \frac{1}{N\sqrt{\pi_k}}\sum_{i=1}^{N} \gamma_i(k)(\frac{\mathbf{x}_i - \boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k}), \tag{2}$$

$$\mathcal{G}_{\boldsymbol{\sigma},k}^{\mathbf{X}} = \frac{1}{N\sqrt{2\pi_k}}\sum_{i=1}^{N} \gamma_i(k)[\frac{(\mathbf{x}_i - \boldsymbol{\mu}_k)^2}{\boldsymbol{\sigma}_k^2} - 1], \tag{3}$$

where $\gamma_i(k)$ is the probability that the i-th local descriptor $\mathbf{x}_i$ belongs to the k-th Gaussian model, which is defined as,

$$\gamma_i(k) = \frac{\pi_k \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j)}, \tag{4}$$

in which $\mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k)$ is the probability of $\mathbf{x}_i$ based on the Gaussian distribution of the k-th Gaussian model. Assuming that the dimension of local descriptors is d, then the dimension of the k-th component of Fisher vectors corresponding to the k-th Gaussian model is $2d$ by concatenating (2) and (3). So the final Fisher vector is a $2Kd$-dim vector *w.r.t.* a K-component GMM.

4 Domain Adaptive Fisher Vector

In this section, we introduce our Domain Adaptive Fisher Vector (DAFV) method, in which we select the domain invariant components of Fisher vectors by simultaneously learning a transformation matrix and a regression matrix for visual recognition tasks. In order to make the proposed formulation easier to be optimized, we introduce an intermediate variable and relax our formulation, and then develop an effective algorithm to solve the optimization problem.

4.1 Formulation

Suppose we have n_s source domain samples and n_t target domain samples from C categories. Each sample is represented by a $2Kd$-dim Fisher vector, in which d is the dimension of local descriptors and K is the number of Gaussian models in the GMM. Let us denote $\mathbf{X}^s \in \mathcal{R}^{2Kd\times n_s}$ and $\mathbf{X}^t \in \mathcal{R}^{2Kd\times n_t}$ as the features of source domain samples and target domain samples, and $\mathbf{Y} \in \mathcal{Z}^{C\times n_s}$ as the binary label matrix for the source domain samples. In order to select domain invariant components and simultaneously keep discriminative information, we use the transformation matrix $\mathbf{R} \in \mathcal{R}^{m\times 2Kd}$ to project the original Fisher vector to lower dimensional subspace with m being the dimension of transformed features. We employ the group lasso based regularizer [21] $\|\tilde{\mathbf{R}}\|_{2,1}$ to enforce each column of $\tilde{\mathbf{R}}$ to have either all zero weights or multiple nonzero weights, in which $\tilde{\mathbf{R}} \in \mathcal{R}^{2d\times Km}$ is a reshaped matrix of $\mathbf{R}$ by setting each group of $2d$ entries in each row of $\mathbf{R}$ corresponding to one component in the Fisher vector as one column in $\tilde{\mathbf{R}}$. To be exact, we expect to assign nonzero weights to the selected domain invariant components of Fisher vectors and zero weights to the remaining ones.

To ensure the selected components are domain invariant, we tend to minimize the Maximum Mean Discrepancy (MMD) of transformed features between the source domain and the target domain by using an MMD-based [11] regularizer $\|\frac{1}{n_s}\mathbf{R}\mathbf{X}^s\mathbf{1} - \frac{1}{n_t}\mathbf{R}\mathbf{X}^t\mathbf{1}\|^2$, in which $\frac{1}{n_s}\mathbf{R}\mathbf{X}^s\mathbf{1}$ (*resp.*, $\frac{1}{n_t}\mathbf{R}\mathbf{X}^t\mathbf{1}$) is the mean of transformed features from the source (*resp.*, target) domain, so that the data distribution mismatch between two domains can be reduced. Additionally, inspired by [17], we add a constraint $\mathbf{RXHX'R'} = \mathbf{I}$ to maximally preserve the data variance, where $\mathbf{X} = [\mathbf{X}^s, \mathbf{X}^t]$ and $\mathbf{H} = \mathbf{I}_n - \frac{1}{n}\mathbf{1}\mathbf{1}'$ with $n = n_s + n_t$.

By denoting $\mathbf{W} \in \mathcal{R}^{C\times m}$ as the regression matrix, we formulate our method by solving the following regression problem:

$$\min_{\mathbf{W},\mathbf{R}} \frac{1}{2}\|\mathbf{WRX}^s - \mathbf{Y}\|_F^2 + \frac{\gamma}{2}\|\mathbf{W}\|_F^2 + \lambda\|\tilde{\mathbf{R}}\|_{2,1} + \frac{1}{2}\|\frac{1}{n_s}\mathbf{RX}^s\mathbf{1} - \frac{1}{n_t}\mathbf{RX}^t\mathbf{1}\|^2 \quad (5)$$

$$\text{s.t.} \quad \mathbf{RXHX'R'} = \mathbf{I}, \quad (6)$$

in which $\|\mathbf{WRX}^s - \mathbf{Y}\|_F^2$ is the regression error, $\|\mathbf{W}\|_F^2$ is the weight decay regularizer to control the complexity of $\mathbf{W}$, γ and λ are two trade-off parameters.

The problem in (5) is not easy to solve due to the constraint in (6). For ease of optimization, we introduce an intermediate variable $\mathbf{S}$ and promote the coherence between $\mathbf{R}$ and $\mathbf{S}$ by adding a coherent regularizer $\|\mathbf{RS'}\|_F^2$ [36]. With larger $\|\mathbf{RS'}\|_F^2$, $\mathbf{R}$ is more coherent to $\mathbf{S}$. As a result, the proposed formulation after introducing $\mathbf{S}$ becomes,

$$\begin{aligned}\min_{\mathbf{W},\mathbf{R},\mathbf{S}} \quad & \frac{1}{2}\|\mathbf{W}\mathbf{S}\mathbf{X}^s - \mathbf{Y}\|_F^2 + \frac{\gamma}{2}\|\mathbf{W}\|_F^2 + \lambda\|\tilde{\mathbf{S}}\|_{2,1} \\ & + \frac{1}{2}\|\frac{1}{n_s}\mathbf{R}\mathbf{X}^s\mathbf{1} - \frac{1}{n_t}\mathbf{R}\mathbf{X}^t\mathbf{1}\|^2 - \frac{1}{2}\|\mathbf{R}\mathbf{S}'\|_F^2 \qquad (7) \\ \text{s.t.} \quad & \mathbf{R}\mathbf{X}\mathbf{H}\mathbf{X}'\mathbf{R}' = \mathbf{I}. \qquad (8)\end{aligned}$$

By replacing $\mathbf{R}$ in $\|\mathbf{W}\mathbf{R}\mathbf{X}^s - \mathbf{Y}\|_F^2$ and $\|\tilde{\mathbf{R}}\|_{2,1}$ in (5) by $\mathbf{S}$, the subproblem *w.r.t.* $\mathbf{R}$ in (7) can be easily solved by using eigen decomposition, which will be discussed in detail in the next section.

Another problem is that the dimension of Fisher vector is usually very high. Considering high time-complexity operations such as eigen decomposition, the algorithm will become very time-consuming. To accelerate the algorithm and simultaneously capture the semantic information within each category, we partition each Fisher vector into C uncorrelated parts by training a category-specific GMM with a smaller number of Gaussian models based on the training samples within each category. Then, a set of $\mathbf{W}_c$, $\mathbf{R}_c$, and $\mathbf{S}_c$ is learnt for the components of each Fisher vector corresponding to the c-th GMM. As a result, we have totally C sets of $\mathbf{W}_c \in \mathcal{R}^{C\times\bar{m}}$, $\mathbf{R}_c \in \mathcal{R}^{\bar{m}\times 2\bar{K}d}$, and $\mathbf{S}_c \in \mathcal{R}^{\bar{m}\times 2\bar{K}d}$ for $c = 1,\ldots,C$, in which we denote the number of Gaussian models in each category-specific GMM as $\bar{K}$ ($\bar{K} << K$) and the dimension of the transformed features corresponding to each category-specific GMM as $\bar{m}$ ($\bar{m} << m$). Correspondingly, we partition the training (*resp.*, test) features $\mathbf{X}^s$ (*resp.*, $\mathbf{X}^t$) into $\mathbf{X}_c^s \in \mathcal{R}^{2\bar{K}d\times n_s}$'s (*resp.*, $\mathbf{X}_c^t \in \mathcal{R}^{2\bar{K}d\times n_t}$'s) with each obtained based on the c-th GMM, and denote $\mathbf{X}_c = [\mathbf{X}_c^s, \mathbf{X}_c^t]$. In fact, supervised learning for GMM (*i.e.*, train one GMM per category) has been studied in [37] and proved to be able to preserve the useful discriminative information. To this end, we can relax the problem in (7) as,

$$\begin{aligned}\min_{\mathbf{W}_c,\mathbf{R}_c,\mathbf{S}_c} \quad & \frac{1}{2}\|\sum_{c=1}^{C}\mathbf{W}_c\mathbf{S}_c\mathbf{X}_c^s - \mathbf{Y}\|_F^2 + \frac{\gamma}{2}\sum_{c=1}^{C}\|\mathbf{W}_c\|_F^2 + \lambda\sum_{c=1}^{C}\|\tilde{\mathbf{S}}_c\|_{2,1} \\ & + \frac{1}{2}\sum_{c=1}^{C}\|\frac{1}{n_s}\mathbf{R}_c\mathbf{X}_c^s\mathbf{1} - \frac{1}{n_t}\mathbf{R}_c\mathbf{X}_c^t\mathbf{1}\|^2 - \frac{1}{2}\sum_{c=1}^{C}\|\mathbf{R}_c\mathbf{S}_c'\|_F^2 \qquad (9) \\ \text{s.t.} \quad & \mathbf{R}_c\mathbf{X}_c\mathbf{H}\mathbf{X}_c'\mathbf{R}_c' = \mathbf{I}, \quad \forall c. \qquad (10)\end{aligned}$$

By partitioning a Fisher vector into C uncorrelated parts, we can solve C small-scale subproblems instead of a large-scale problem, which is more efficient. Considering the tradeoff between efficiency and effectiveness, we set $\bar{K}$ as 8 and $\bar{m}$ as 1000 in our experiments. Moreover, another benefit of replacing $\|\tilde{\mathbf{S}}\|_{2,1}$ with $\|\tilde{\mathbf{S}}_c\|_{2,1}$ is that we can guarantee at least one Gaussian model is selected from each category-specific GMM, which ensures capturing the semantic information over all categories. Next, we will discuss how to solve the problem in (9).

4.2 Optimization

We solve the problem in (9) by using an alternative optimization approach. Specifically, we alteratively update three sets of variables $\mathbf{W}_c$'s, $\mathbf{S}_c$'s, and $\mathbf{R}_c$'s until the objective value of (9) converges.

Update $\mathbf{W}_c$ when fixing $\mathbf{R}_c$ and $\mathbf{S}_c$: When fixing $\mathbf{R}_c$'s and $\mathbf{S}_c$'s, the problem in (9) reduces to:

$$\min_{\mathbf{W}_c} \frac{1}{2}\|\sum_{c=1}^{C}\mathbf{W}_c\mathbf{S}_c\mathbf{X}_c^s - \mathbf{Y}\|_F^2 + \frac{\gamma}{2}\sum_{c=1}^{C}\|\mathbf{W}_c\|_F^2 \tag{11}$$

By setting the derivative of (11) *w.r.t.* each $\mathbf{W}_c$ to $\mathbf{0}$, we can derive the close-form solution for each $\mathbf{W}_c$ as,

$$\mathbf{W}_c = (\mathbf{Y} - \sum_{\tilde{c}=1,\tilde{c}\neq c}^{C}\mathbf{W}_{\tilde{c}}\mathbf{S}_{\tilde{c}}\mathbf{X}_{\tilde{c}}^s)\mathbf{X}_c^{s\prime}\mathbf{S}_c'(\mathbf{S}_c\mathbf{X}_c^s\mathbf{X}_c^{s\prime}\mathbf{S}_c' + \gamma\mathbf{I})^{-1}. \tag{12}$$

We calculate each $\mathbf{W}_c$ when fixing all the other $\mathbf{W}_{\tilde{c}}$ for $\tilde{c} \neq c$ and repeat this process iteratively until the objective value of (11) converges.

Update $\mathbf{R}_c$ when fixing $\mathbf{W}_c$ and $\mathbf{S}_c$: When fixing $\mathbf{W}_c$'s and $\mathbf{S}_c$'s, the problem in (9) can be separated into C independent subproblems with one for each $\mathbf{R}_c$. For ease of optimization, we rewrite the subproblem *w.r.t.* each $\mathbf{R}_c$ by using trace norm as follows,

$$\min_{\mathbf{R}_c} \ \frac{1}{2}\mathrm{tr}(\mathbf{R}_c\mathbf{X}_c\mathbf{L}\mathbf{X}_c'\mathbf{R}_c') - \frac{1}{2}\mathrm{tr}(\mathbf{R}_c\mathbf{S}_c'\mathbf{S}_c\mathbf{R}_c') \tag{13}$$

$$\text{s.t. } \ \mathbf{R}_c\mathbf{X}_c\mathbf{H}\mathbf{X}_c'\mathbf{R}_c' = \mathbf{I}, \tag{14}$$

where $\mathbf{L}$ is an indicator matrix, in which $L_{ij} = \frac{1}{n_s^2}$ if $i \leq n_s$ and $j \leq n_s$; else $L_{ij} = \frac{1}{n_t^2}$ if $i > n_s$ and $j > n_s$; otherwise, $L_{ij} = -\frac{1}{n_s n_t}$.

By introducing a symmetric matrix $\mathbf{Z}_c$ containing the Lagrangian multipliers for the constraints in (14), we obtain the Lagrangian form of (13) as,

$$\mathcal{L}_{\mathbf{R}_c,\mathbf{Z}_c} = \mathrm{tr}(\mathbf{R}_c(\frac{1}{2}\mathbf{X}_c\mathbf{L}\mathbf{X}_c' - \frac{1}{2}\mathbf{S}_c'\mathbf{S}_c)\mathbf{R}_c') - \mathrm{tr}((\mathbf{R}_c\mathbf{X}_c\mathbf{H}\mathbf{X}_c'\mathbf{R}_c' - \mathbf{I})\mathbf{Z}_c). \tag{15}$$

By setting the derivative of (15) *w.r.t.* $\mathbf{R}_c$ to $\mathbf{0}$, we arrive at

$$\mathbf{R}_c(\mathbf{X}_c\mathbf{L}\mathbf{X}_c' - \mathbf{S}_c'\mathbf{S}_c) = 2\mathbf{Z}_c\mathbf{R}_c\mathbf{X}_c\mathbf{H}\mathbf{X}_c'. \tag{16}$$

Multiplying both sides on the right by $\mathbf{R}_c'$, we obtain the solution *w.r.t.* $\mathbf{Z}_c$ as follows,

$$\mathbf{Z}_c = \frac{1}{2}(\mathbf{R}_c(\mathbf{X}_c\mathbf{L}\mathbf{X}_c' - \mathbf{S}_c'\mathbf{S}_c)\mathbf{R}_c')(\mathbf{R}_c\mathbf{X}_c\mathbf{H}\mathbf{X}_c'\mathbf{R}_c')^{-1}. \tag{17}$$

By substituting (17) back into (15) followed by some simplifications, we derive the dual form of (13) as,

$$\max_{\mathbf{R}_c} \frac{1}{2}\mathrm{tr}((\mathbf{R}_c\mathbf{X}_c\mathbf{H}\mathbf{X}_c'\mathbf{R}_c')^{-1}(\mathbf{R}_c(\mathbf{X}_c\mathbf{L}\mathbf{X}_c' - \mathbf{S}_c'\mathbf{S}_c)\mathbf{R}_c')) \tag{18}$$

Similar to kernel Fisher discriminant analysis [38], the problem in (18) can be solved by eigen decomposition and the rows of $\mathbf{R}_c$ are the $\bar{m}$ leading eigen vectors of $(\mathbf{X}_c\mathbf{H}\mathbf{X}_c')^{-1}(\mathbf{X}_c\mathbf{L}\mathbf{X}_c' - \mathbf{S}_c'\mathbf{S}_c)$.

Update $\mathbf{S}_c$ when fixing $\mathbf{R}_c$ and $\mathbf{W}_c$: When fixing $\mathbf{R}_c$'s and $\mathbf{W}_c$'s, the problem in (9) reduces to the following problem:

$$\min_{\mathbf{S}_c} \frac{1}{2}\|\sum_{c=1}^{C}\mathbf{W}_c\mathbf{S}_c\mathbf{X}_c^s - \mathbf{Y}\|_F^2 + \lambda\sum_{c=1}^{C}\|\tilde{\mathbf{S}}_c\|_{2,1} - \frac{1}{2}\sum_{c=1}^{C}\|\mathbf{R}_c\mathbf{S}_c'\|_F^2 \tag{19}$$

The optimization problem in (19) is non-convex and thus only local optimum can be reached by using gradient descent algorithm. First, we derive the derivative of each term in (19) *w.r.t.* each $\mathbf{S}_c$ separately.

$$\mathbf{J}_1 = \frac{\partial\frac{1}{2}\|\sum_{c=1}^{C}\mathbf{W}_c\mathbf{S}_c\mathbf{X}_c^s - \mathbf{Y}\|_F^2}{\partial\mathbf{S}_c} = \mathbf{W}_c'(\sum_{c=1}^{C}\mathbf{W}_c\mathbf{S}_c\mathbf{X}_c^s - \mathbf{Y})\mathbf{X}_c^{s\prime}, \tag{20}$$

$$\mathbf{J}_2 = \frac{\partial\lambda\|\tilde{\mathbf{S}}_c\|_{2,1}}{\partial\mathbf{S}_c} = \lambda\mathbf{S}_c \circ \mathbf{D}_c, \tag{21}$$

where $\mathbf{D}_c \in \mathcal{R}^{\bar{m}\times 2\bar{K}d}$ is a matrix, in which each entry D_c^{ij} is set as $\frac{1}{\|\mathbf{S}_c^{ik}\|_2}$ if j belongs to the k-th component, with $\mathbf{S}_c^{i,k}$ denoting the k-th component in the i-th row of $\mathbf{S}_c$.

$$\mathbf{J}_3 = \frac{\partial - \frac{1}{2}\|\mathbf{R}_c\mathbf{S}_c'\|_F^2}{\partial\mathbf{S}_c} = -\mathbf{S}_c\mathbf{R}_c'\mathbf{R}_c. \tag{22}$$

In each iteration, we update each $\mathbf{S}_c$ when fixing all the other $\mathbf{S}_{\tilde{c}}$'s for $\tilde{c} \neq c$ by using the following equation:

$$\mathbf{S}_c \leftarrow \mathbf{S}_c - \eta(\mathbf{J}_1 + \mathbf{J}_2 + \mathbf{J}_3), \tag{23}$$

where η is the learning rate, which is empirically fixed as 0.0001 in our experiments. We repeat this process iteratively until the objective value of (19) converges. The whole algorithm is summarized in Algorithm 1. The objective value of (9) monotonically decreases as the number of iterations increases and usually converges within 20 iterations in our experiments.

In the testing stage, for each test sample $\mathbf{x}^t$ which contains the features $\mathbf{x}_c^t$'s obtained based on each category-specific GMM, we use $\sum_{c=1}^{C}\mathbf{W}_c\mathbf{S}_c\mathbf{x}_c^t$ to obtain the regression values and assign this test sample to the category corresponding to the maximum regression value.

Algorithm 1. Domain Adaptive Fisher Vector (DAFV) Algorithm

```
Input: $\mathbf{X}_c^s, \mathbf{X}_c^t, \mathbf{Y}, \lambda, \gamma$
Initialize $\mathbf{S}_c$ as the PCA projection matrix on $\mathbf{X}_c$.
repeat
    repeat
        For c=1,...,C, update $\mathbf{W}_c$ using (12).
    until The objective of (11) converges.
    For c=1,...,C, update $\mathbf{R}_c$ by solving (18).
    repeat
        For c=1,...,C, update $\mathbf{S}_c$ using (23).
    until The objective of (19) converges.
until The objective of (9) converges.
Output: $\mathbf{W}_c$, $\mathbf{S}_c$.
```

5 Experiments

In this section, we demonstrate the effectiveness of our Domain Adaptive Fisher Vector (DAFV) approach for object recognition and human action recognition by conducting extensive experiments on two benchmark datasets.

5.1 Object Recognition

Experimental Settings: We use Bing-Caltech256 [39] dataset, which is commonly used to evaluate domain adaption methods for object recognition. Bing-Caltech256 dataset consists of the images from Caltech256 dataset and the images from Bing search engine distributed in 256 categories. Generally, Bing is treated as the source domain and Caltech-256 is treated as the target domain, because Bing images are collected by the search engine without having ground-truth labels and thus not appropriate for being used as test set. Following the setting in [40], we use the first 20 categories and set the number of source (*resp.*, target) domain examples per category to be 50 (*resp.*, 25) based on the train/test split provided in [39].

In order to generate local descriptors for each image, we first use selection search [41] to generate object proposals. Then, we use the output of the 6-th layer of AlexNet [42] as the 4096-dim feature for each proposal with the pretrained model in [43]. After reducing the dimension of proposal features to 200 by using Principle Component Analysis (PCA), we use the proposals from the source domain within each category to train an 8-component Gaussian Mixture Model (GMM), which leads to a total of 160 components for all categories. Finally, we encode each image, which is a bag of 200-dim proposal features, as a $64,000$-dim Fisher vector based on the trained GMMs.

Baselines: We compare our DAFV method with two sets of baselines: domain adaptation baselines and GMM based baselines. We also include Regularized Least Square (RLS) as a baseline. For domain adaptation baselines, we compare our method with feature-based methods GFK [13], SGF [14], SA [15], DIP [16], TCA [17], LSSA [18], CORAL [28], the SVM-based method DASVM [29], the instance reweighting method KMM [11], the dictionary learning method SDDL [19], and the low-rank based method LTSL [20]. Note that for feature-based methods [13–18], we first obtain the transformed features by employing their methods suggested in the original papers [13–18] and then use the transformed features as input features for RLS.

For GMM based baselines AGMM [8] and EM_RGMM [10], we use different approaches to obtain GMMs, which is explained as follows,

– AGMM [8]: We first train a 160-component GMM by using proposals from the source domain, and then adapt this GMM using the proposals from the target domain. Based on the GMM on the source domain and the adapted GMM, we extract two sets of Fisher vectors for all images from both domains. Based on these two sets of Fisher vectors, we train regressors and obtain the regression values of test images separately, and finally use the average fusion of two sets of regression values for prediction.
– EM_RGMM [10]: We train two 160-component GMMs based on the proposals from the source domain and the target domain, separately. Then, we calculate the interpolated GMM between the two GMMs. Based on the interpolated GMM, we extract Fisher vectors for all images from both domains. Finally, we train regressors and predict the test images based on the extracted Fisher vectors.

Table 1. Accuracies (%) of RLS and GMM based baselines, as well as our DAFV method and its two special cases for object recognition. The best result is denoted in boldface

RLS	AGMM	EM_RGMM	DAFV_sim1	DAFV_sim2	DAFV
73.2	76.8	77.4	75.4	77.8	**79.4**

Table 2. Accuracies (%) of domain adaptation baselines and our DAFV method for object recognition. The best result is denoted in boldface

KMM	DASVM	GFK	SGF	SA	DIP	TCA	LSSA	SDDL	LTSL	CORAL	DAFV
73.6	75.8	73.6	74.4	74.2	71.8	74.8	77.8	62.4	77.6	75.20	**79.4**

Moreover, in order to validate our MMD-based regularizer and group lasso based regularizer, we compare our method with its two simplified versions. Specifically, we remove the group lasso based regularizer $\sum_{c=1}^{C} \|\tilde{\mathbf{S}}_c\|_{2,1}$ in (9) by setting the parameter λ as 0 and refer to this special case as DAFV_sim2. Based on DAFV_sim2, we further remove the MMD-based regularizer $\|\frac{1}{n_s}\mathbf{R}_c\mathbf{X}_c^s\mathbf{1} - \frac{1}{n_t}\mathbf{R}_c\mathbf{X}_c^t\mathbf{1}\|^2$ and denote this special case as DAFV_sim1.

We use accuracy for performance evaluation. Two trade-off parameters γ and λ in (9) are empirically set as 1000 and 10 for our DAFV method. For the baseline methods, we choose their optimal parameters based on their accuracies on the test dataset.

Experimental Results: We report the results of RLS, the GMM based baselines, and our DAFV method including its two special cases in Table 1, from which we observe that AGMM and EM_RGMM achieve better results than RLS, suggesting the benefits of adapting or interpolating GMMs. We also observe that our DAFV method outperforms DAFV_sim2, which validates the effectiveness of selecting some components of Fisher vectors by using group lasso based regularizer. Additionally, DAFV_sim2 outperforms DAFV_sim1, which validates our MMD based regularizer. Finally, our DAFV method outperforms the GMM based baselines, which shows its effectiveness on reducing domain distribution mismatch between the source domain and the target domain.

Moreover, we report the results of domain adaptation baselines in Table 2 and also include the result of our DAFV method for comparison. From Table 2, we observe that the domain adaptation baselines are generally better than RLS reported in Table 1. The results validate the effectiveness of employing different strategies to address the domain shift issue. However, all the domain adaptation baselines are worse than our DAFV method. One possible explanation is that we select the domain invariant components of Fisher vectors, which is designed for Fisher vectors.

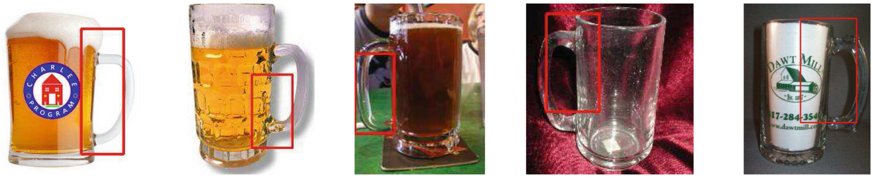

Fig. 1. The top object proposals belonging to the selected Gaussian model for the “beer-mug” category from the Bing dataset

Discussion on Domain Invariant Components: As discussed in Sect. 1, the motivation of our DAFV method is that each Gaussian model in the GMM represents the data distribution of a cluster of local descriptors and corresponds to one component in the encoded Fisher vector. Assuming that there exist some Gaussian models representing common distribution shared by both source and target domain, the corresponding components of Fisher vectors should be more domain invariant. The benefit of selecting domain invariant components has been demonstrated in Tables 1 and 2, and now we provide some intuitive examples to illustrate the domain invariant components.

First, recall that we train C category-specific GMMs and $\mathbf{S}_c \in \mathcal{R}^{\bar{m}\times 2\bar{K}d}$ is the transformation matrix corresponding to the c-th GMM. For the c-th category, we compute the L_2 norm for each component in each row of $\mathbf{S}_c$, which corresponds to one Gaussian model in the c-th GMM. Then, we sum the computed values over different rows and choose the component with the maximum value, which corresponds to the selected Gaussian model in the c-th GMM. Because there are probabilities $\gamma_i(k)$'s that the i-th proposal belongs to the k-th Gaussian model (see Sect. 3) when training a GMM, we can easily pick out the top proposals that belong to the cluster corresponding to the selected Gaussian model. Let us take the "beer-mug" category as an example to show the top proposals for the selected Gaussian model in Fig. 1, from which we have an interesting observation that the proposals are all near the handle of beer mug. We conjecture beer mugs from different domains are quite different in shape, color, and pattern of body regions, but the handle regions generally look similar as illustrated in Fig. 1. Intuitively, the handle regions can be used to discriminate beer mugs against the other categories but are less variant across different domains. So the components of Fisher vectors corresponding to the selected Gaussian models are assigned larger weights, which is helpful for improving the performance of object recognition.

5.2 Human Action Recognition

Experimental Settings: We use the ACT4^2 [44] dataset for human action recognition. The ACT4^2 dataset contains videos from 14 categories of human actions, which are captured from 4 camera viewpoints. Following [44], we use a subset with 2648 RGB videos from all 4 viewpoints. We treat one view as the

Table 3. Accuracies (%) of RLS and GMM based baselines, as well as our DAFV method and its two special cases for human action recognition. The best results on each setting are denoted in boldface

Setting	RLS	AGMM	EM_RGMM	DAFV_sim1	DAFV_sim2	DAFV
1->2	69.94	72.36	73.72	71.00	72.96	**74.92**
1->3	44.11	46.07	46.22	45.02	46.68	**48.49**
1->4	77.64	80.21	80.06	81.27	82.33	**83.99**
2->1	74.17	77.95	74.02	77.04	77.64	**79.61**
2->3	67.37	67.52	67.82	69.94	71.00	**72.96**
2->4	60.88	61.03	61.18	60.57	62.24	**63.90**
3->1	52.87	47.89	51.96	51.21	52.87	**55.74**
3->2	66.92	66.92	67.07	69.18	69.94	**71.90**
3->4	40.03	41.69	41.99	41.69	43.20	**45.47**
4->1	71.75	73.72	72.21	68.73	75.98	**76.13**
4->2	46.37	52.27	52.11	49.40	51.96	**53.92**
4->3	37.31	38.97	36.71	38.52	40.03	**41.69**
Avg	59.11	60.55	60.42	60.30	62.24	**64.06**

Table 4. Accuracies (%) of domain adaptation baselines and our DAFV method for human action recognition. The best results on each setting are denoted in boldface

Setting	KMM	DASVM	GFK	SGF	SA	DIP	TCA	LSSA	SDDL	LTSL	CORAL	DAFV
1->2	67.67	59.52	73.11	66.16	72.96	72.21	72.81	73.56	72.96	71.75	72.69	**74.92**
1->3	45.62	35.65	46.37	45.02	45.92	46.37	46.53	44.11	45.02	45.17	46.37	**48.49**
1->4	79.91	74.17	81.72	78.85	80.97	80.51	82.93	82.33	79.00	81.72	81.72	**83.99**
2->1	76.74	68.88	77.95	70.85	75.98	75.38	**79.76**	68.88	75.98	75.68	75.68	79.61
2->3	69.94	55.29	70.54	66.62	69.79	71.60	69.49	65.41	69.94	68.73	71.90	**72.96**
2->4	61.33	56.34	61.48	59.06	62.08	62.84	61.78	62.08	61.33	61.33	61.63	**63.90**
3->1	54.98	48.94	53.78	47.73	54.08	54.53	54.68	50.45	53.47	54.08	53.47	**55.74**
3->2	70.54	62.08	69.94	69.79	67.07	71.00	67.67	64.20	68.88	67.82	67.67	**71.90**
3->4	41.39	32.33	42.60	41.09	42.45	43.20	43.35	40.94	36.40	43.96	43.81	**45.47**
4->1	74.62	67.98	73.11	74.17	73.87	73.87	73.26	66.01	74.47	72.36	73.72	**76.13**
4->2	**54.83**	46.37	49.24	53.02	53.32	51.66	49.85	52.11	47.43	51.66	52.27	53.92
4->3	34.29	36.40	40.03	39.43	38.97	40.33	39.27	39.73	37.61	38.97	40.48	**41.69**
Avg	60.99	53.66	61.66	59.32	61.46	61.96	61.78	59.15	60.21	61.10	61.81	**64.06**

source domain and another different view as the target domain, which results in totally 12 settings.

Following [6], we use the source codes provided in [6] to extract four types of Improved Dense Trajectory (IDT) descriptors (i.e., 30-dim trajectories, 96-dim HOG, 108-dim HOF, and 192-dim MBH). Following [6], we first reduce the dimension of descriptors by a factor of two using PCA. Then, we use the descriptors from the videos in the source domain within each category to train an 8-component GMM, which leads to totally 112 components for all categories. Finally, we encode each video, which is a bag of 213-dim IDT descriptors, as a 47712-dim Fisher vector based on the trained GMMs.

Baselines: We compare our DAFV method with the same baselines as discussed in Sect. 5.1. The only difference is that we train 112-component GMMs for AGMM and EM_RGMM. For the human action recognition task, accuracy is still used for performance evaluation. Our DAFV method employs the same parameters as used for object recognition while optimal parameters of the baseline methods are chosen according to their accuracies on the test dataset.

Experimental Results: We report the experimental results of RLS and GMM based baselines, as well as our DAFV method and its two special cases on 12 settings in Table 3. From the results, we can draw similar conclusions as those for object recognition in Sect. 5.1. In particular, the comparisons among our DAFV method and its two special cases clearly demonstrate the effectiveness of our group lasso based regularizer and the MMD-based regularizer. Moreover, our DAFV method is better than the GMM based baselines on all settings. The results again demonstrate that the recognition performance can be improved by reducing domain distribution mismatch.

Table 4 shows the results of domain adaptation baselines. It can be seen that the average accuracies of the domain adaptation baselines are better than that of RLS reported in Table 3 except DASVM, which indicates the advantage of coping with domain difference by using various methods. While TCA (*resp.*, KMM) is better than our DAFV method on the setting 2->1 (*resp.*, 4->2), our method achieves the best results on 10 out of 12 settings. Moreover, in terms of the average accuracy over 12 settings, our DAFV method is the best, which again demonstrates it is helpful to address the domain shift issue by selecting domain invariant components of Fisher vectors.

6 Conclusion

In this paper, we have proposed a domain adaptation method named Domain Adaptive Fisher Vector (DAFV), which is designed for Fisher vectors. Based on the assumption that some Gaussian models in the GMM can better capture the common data distribution between the source domain and the target domain, our DAFV method is designed to select the domain invariant components of Fisher vectors corresponding to the common Gaussian models and simultaneously solve a regression problem. The effectiveness of our DAFV method for visual recognition has been demonstrated by extensive experiments.

Acknowledgement. This research was partially carried out at the Rapid-Rich Object Search (ROSE) Laboratory, Nanyang Technological University, Singapore. The ROSE Laboratory is supported by the National Research Foundation, Prime Ministers Office, Singapore, under its IDM Futures Funding Initiative and administered by the Interactive and Digital Media Programme Office.

References

1. Jaakkola, T.S., Haussler, D., et al.: Exploiting generative models in discriminative classifiers. In: NIPS (1999)
2. Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 143–156. Springer, Heidelberg (2010)
3. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep Fisher networks for large-scale image classification. In: NIPS (2013)
4. Wei, Y., Xia, W., Huang, J., Ni, B., Dong, J., Zhao, Y., Yan, S.: CNN: Single-label to multi-label (2014). arXiv preprint arXiv:1406.5726
5. Uricchio, T., Bertini, M., Seidenari, L., Bimbo, A.: Fisher encoded convolutional bag-of-windows for efficient image retrieval and social image tagging. In: ICCV (2015)
6. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: ICCV (2013)
7. Peng, X., Zou, C., Qiao, Y., Peng, Q.: Action recognition with stacked fisher vectors. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part V. LNCS, vol. 8693, pp. 581–595. Springer, Heidelberg (2014)
8. Dixit, M., Rasiwasia, N., Vasconcelos, N.: Adapted Gaussian models for image classification. In: CVPR (2011)
9. Torralba, A., Efros, A.A.: Unbiased look at dataset bias. In: CVPR (2011)

10. Kim, H.J., Adluru, N., Banerjee, M., Vemuri, B.C., Singh, V.: Interpolation on the manifold of K component GMMs. In: ICCV (2015)
11. Huang, J., Smola, A., Gretton, A., Borgwardt, K., Scholkopf, B.: Correcting sample selection bias by unlabeled data. In: NIPS (2007)
12. Bruzzone, L., Marconcini, M.: Domain adaptation problems: A DASVM classification technique and a circular validation strategy. T-PAMI **32**(5), 770–787 (2010)
13. Gong, B., Shi, Y., Sha, F., Grauman, K.: Geodesic flow kernel for unsupervised domain adaptation. In: CVPR (2012)
14. Gopalan, R., Li, R., Chellappa, R.: Domain adaptation for object recognition: an unsupervised approach. In: ICCV (2011)
15. Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T.: Unsupervised visual domain adaptation using subspace alignment. In: ICCV (2013)
16. Baktashmotlagh, M., Harandi, M.T., Lovell, B.C., Salzmann, M.: Unsupervised domain adaptation by domain invariant projection. In: ICCV (2013)
17. Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q.: Domain adaptation via transfer component analysis. T-NN **22**(2), 199–210 (2011)
18. Aljundi, R., Emonet, R., Muselet, D., Sebban, M.: Landmarks-based kernelized subspace alignment for unsupervised domain adaptation. In: CVPR (2015)
19. Shekhar, S., Patel, V., Nguyen, H., Chellappa, R.: Generalized domain-adaptive dictionaries. In: CVPR (2013)
20. Shao, M., Kit, D., Fu, Y.: Generalized transfer subspace learning through low-rank constraint. IJCV **109**(1–2), 74–93 (2014)
21. Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. J. Royal Stat. Soc. Ser. B (Statistical Methodology) **68**(1), 49–67 (2006)
22. Perronnin, F., Dance, C.: Fisher kernels on visual vocabularies for image categorization. In: CVPR (2007)
23. Liu, L., Shen, C., Wang, L., van den Hengel, A., Wang, C.: Encoding high dimensional local features by sparse coding based Fisher vectors. In: NIPS (2014)
24. Gong, Y., Wang, L., Guo, R., Lazebnik, S.: Multi-scale orderless pooling of deep convolutional activation features. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part VII. LNCS, vol. 8695, pp. 392–407. Springer, Heidelberg (2014)
25. Oneata, D., Verbeek, J., Schmid, C.: Action and event recognition with Fisher vectors on a compact feature set. In: ICCV (2013)
26. Caseiro, R., Henriques, J.F., Martins, P., Batista, J.: Beyond the shortest path: Unsupervised domain adaptation by sampling subspaces along the spline flow. In: CVPR (2015)
27. Kulis, B., Saenko, K., Darrell, T.: What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In: CVPR (2011)
28. Sun, B., Feng, J., Saenko, K.: Return of frustratingly easy domain adaptation. In: AAAI (2016)
29. Duan, L., Tsang, I.W., Xu, D.: Domain transfer multiple kernel learning. T-PAMI **34**, 465–479 (2012)
30. Duan, L., Xu, D., Tsang, I.W., Luo, J.: Visual event recognition in videos by learning from web data. T-PAMI **34**, 1667–1680 (2012)
31. Li, W., Niu, L., Xu, D.: Exploiting privileged information from web data for image categorization. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part V. LNCS, vol. 8693, pp. 437–452. Springer, Heidelberg (2014)
32. Niu, L., Li, W., Xu, D.: Exploiting privileged information from web data for action and event recognition. In: IJCAI (2016)

33. Jhuo, I.H., Liu, D., Lee, D., Chang, S.F., et al.: Robust visual domain adaptation with low-rank reconstruction. In: CVPR (2012)
34. Zhou, X., Cui, N., Li, Z., Liang, F., Huang, T.S.: Hierarchical Gaussianization for image classification. In: ICCV (2009)
35. Fe-Fei, L., Fergus, R., Perona, P.: A bayesian approach to unsupervised one-shot learning of object categories. In: ICCV (2003)
36. Ramirez, I., Sprechmann, P., Sapiro, G.: Classification and clustering via dictionary learning with structured incoherence and shared features. In: CVPR (2010)
37. Farquhar, J., Szedmak, S., Meng, H., Shawe-Taylor, J.: Improving bag-of-keypoints image categorisation: Generative models and pdf-kernels. Technical report, University of Southampton (2005)
38. Müller, K.R., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B.: An introduction to kernel-based learning algorithms. T-NN **12**(2), 181–201 (2001)
39. Bergamo, A., Torresani, L.: Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach. In: NIPS (2010)
40. Hoffman, J., Rodner, E., Donahue, J., Darrell, T., Saenko, K.: Efficient learning of domain-invariant image representations. In: ICLR (2013)
41. Uijlings, J.R., van de Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. IJCV **104**(2), 154–171 (2013)
42. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS, pp. 1097–1105 (2012)
43. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding (2014). arXiv preprint arXiv:1408.5093
44. Cheng, Z., Qin, L., Ye, Y., Huang, Q., Tian, Q.: Human daily action analysis with multi-view and color-depth data. In: Fusiello, A., Murino, V., Cucchiara, R. (eds.) ECCV 2012. LNCS, vol. 7584, pp. 52–61. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33868-7_6