Deep Transfer Metric Learning

Junlin Hu[1], Jiwen Lu[2*], Yap-Peng Tan[1]
[1]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
[2]Advanced Digital Sciences Center, Singapore
jhu007@e.ntu.edu.sg, jiwen.lu@adsc.com.sg, eyptan@ntu.edu.sg

Abstract

Conventional metric learning methods usually assume that the training and test samples are captured in similar scenarios so that their distributions are assumed to be the same. This assumption doesn't hold in many real visual recognition applications, especially when samples are captured across different datasets. In this paper, we propose a new deep transfer metric learning (DTML) method to learn a set of hierarchical nonlinear transformations for cross-domain visual recognition by transferring discriminative knowledge from the labeled source domain to the unlabeled target domain. Specifically, our DTML learns a deep metric network by maximizing the inter-class variations and minimizing the intra-class variations, and minimizing the distribution divergence between the source domain and the target domain at the top layer of the network. To better exploit the discriminative information from the source domain, we further develop a deeply supervised transfer metric learning (DSTML) method by including an additional objective on DTML where the output of both the hidden layers and the top layer are optimized jointly. Experimental results on cross-dataset face verification and person re-identification validate the effectiveness of the proposed methods.

1. Introduction

How to design a good similarity function plays an important role in many computer vision and pattern recognition tasks. Generally, the optimal similarity function for a given vision problem is task-specific because the underlying data distributions for different tasks are usually different. Recent advances in machine learning have shown that learning a distance metric directly from a set of training examples can usually achieve proposing performance than hand-crafted distance metrics [9, 32, 33]. In recent years, a variety of metric learning algorithms have been proposed in the literature [9, 14, 16, 23, 25, 32, 33], and some of them have successfully applied in visual analysis applications such as face recognition [14, 16], image classification [9, 32], human activity recognition [31], person re-identification [19] and visual search [26].

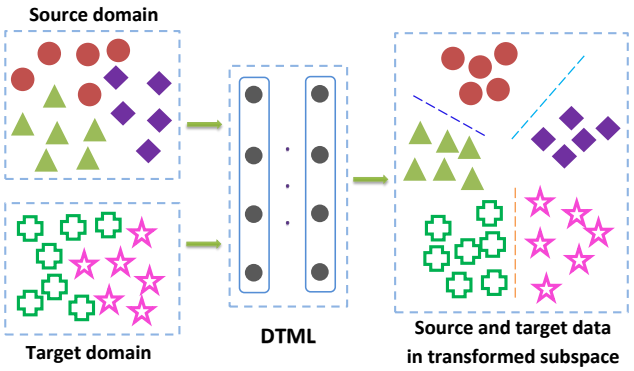

Figure 1. The basic idea of the proposed DTML method. For each sample in the training sets from the source domain and the target domain, we pass it to the developed deep neural network. We enforce two constraints on the outputs of all training samples at the top of the network: 1) the inter-class variations are maximized and the intra-class variations are minimized, and 2) the distribution divergence between the source domain and the target domain at the top layer of the network is minimized.

Existing metric learning methods can be mainly classified into two categories: unsupervised and supervised. For the first category, a low-dimensional subspace or manifold is learned to preserve the geometrical information of the samples. For the second category, a discriminative distance metric is learned to maximize the separability of samples from different classes. Since the label information of training samples is used, supervised metric learning methods are more suitable for the recognition task.

While many supervised metric learning algorithms have been presented in recent years, there are still two shortcomings of these methods: 1) most of them usually seek a single linear distance to transform sample into a linear feature space, so that the nonlinear relationship of samples cannot be well exploited. Even if the kernel trick [36] can be em-

*Corresponding author.

ployed to addressed the nonlinearity issue, these methods still suffer from the scalability problem because they cannot obtain the explicit nonlinear mapping functions; 2) most of them assume that the training and test samples are captured in similar scenarios so that their distributions are assumed to be the same. This assumption doesn't hold in many real visual recognition applications, especially when samples are captured across different datasets.

To this end, in this work, we propose a new deep transfer metric learning (DTML) method for cross-dataset visual recognition. Figure 1 illustrates the basic idea of the proposed method. Our method learns a set of hierarchical nonlinear transformations by transferring discriminative knowledge from the labeled source domain to the unlabeled target domain, under which the inter-class variations are maximized and the intra-class variations are minimized, and the distribution divergence between the source domain and the target domain at the top layer of the network is minimized, simultaneously. To better exploit the discriminative information from the source domain, we further develop a deeply supervised transfer metric learning (DSTML) method by including an additional objective on DTML where the output of both the hidden layers and the top layer are optimized jointly. Experimental results on cross-dataset face verification and person re-identification demonstrate the effectiveness of the proposed methods.

2. Related Work

Deep Learning: In recent years, deep learning has attracted much attention in computer vision and machine learning due to its superb performance in various tasks. Generally, deep learning aims to learn hierarchical feature representations directly from raw data. Recent advances have shown that deep learning have been successfully applied to many visual tasks such as image classification [10, 20], object detection [29], action recognition [21], and face recognition [17, 30]. Many deep learning models have been proposed in recent years, and representative methods include deep convolutional neural networks [20], deep neural networks [4], deep stacked auto-encoder [21], deep belief networks [15], and deeply-supervised nets [22]. However, most of them aim to learn feature representations via deep model rather than similarity measure. More recently, deep learning has also been used in metric learning, and several metric learning methods have been proposed. For example, Cai *et al*. [5] introduced a nonlinear metric learning method using the stacked independent subspace analysis. Hu *et al*. [16] proposed a discriminative deep metric learning method which employs a conventional neural network by enforcing a large margin criterion at the top layer of the network. While these methods have achieved reasonably good performance, they assume that the training and test samples are captured in the same environments, which is not always satisfied in many real applications. In this work, we proposed a deep transfer metric learning approach by learning a deep metric network and considering the distribution difference between the source domain and the target domain.

Transfer Learning: Transfer learning aims to address the problem when the distribution of the training data from the source domain is different from that of the target domain. Over the past decades, a variety of transfer learning algorithms [28] have been proposed and they can be mainly categorized into two classes: instance-based [8] and feature-based [2]. For the first class, different weights are learned to rank the training samples in the source domain for better learning in the target domain. For the second class, a common feature space is usually learned which can transfer the information learned from the source domain to the target domain. In recent years, several transfer learning techniques have been presented and representative methods include domain transfer support vector machine [11], transfer dimensionality reduction [27], and transfer metric learning [37, 38]. While some proposing results can be obtained by these transfer learning methods, most of them only consider minimizing the distribution difference between the source domain and the target domain by using linear mappings or the kernel trick, which are not effective enough to transfer the knowledge if the distribution difference is large and the transfer functions are usually not explicitly obtained. In this work, we borrow the idea of deep learning and propose a deep transfer metric learning method by learning a discriminative distance network with some information transferred from the source domain.

3. DTML

In this section, we first introduce the notation used in this work. Then, we present the deep metric learning framework. Lastly, we present the proposed deep transfer metric learning method.

3.1. Notation

Let $\mathcal{X}_s = \{(\mathbf{x}_{si}, y_{si})|i = 1, 2, \cdots, N_s\}$ be the training set in the source domain, which contains N_s examples, where $\mathbf{x}_{si} \in \mathbb{R}^d$ is a d-dimensional feature vector, $y_{si} \in \{1, 2, \cdots, C_s\}$ is the label of $\mathbf{x}_{si}$, and C_s is the number of classes. Similarly, we denote $\mathcal{X}_t = \{(\mathbf{x}_{ti}, y_{ti})|i = 1, 2, \cdots, N_t\}$ be the training samples in the target domain, where N_t is the number of samples in the set, y_{ti} is the label of $\mathbf{x}_{ti}$. Let $\mathcal{X} = \{(\mathbf{x}_i, y_i)|i = 1, 2, \cdots, N\}$ be the labeled training set, which contains samples either only from the source domain $\mathcal{X}_s$ or from both the source domain $\mathcal{X}_s$ and the target domain $\mathcal{X}_t$. In our experiments, we consider a challenging case where the labeled training set $\mathcal{X}$ is only sampled from the source domain $\mathcal{X}_s$ (*i.e.*, $\mathcal{X} = \mathcal{X}_s$) and no labeled training set is obtained from the target domain.

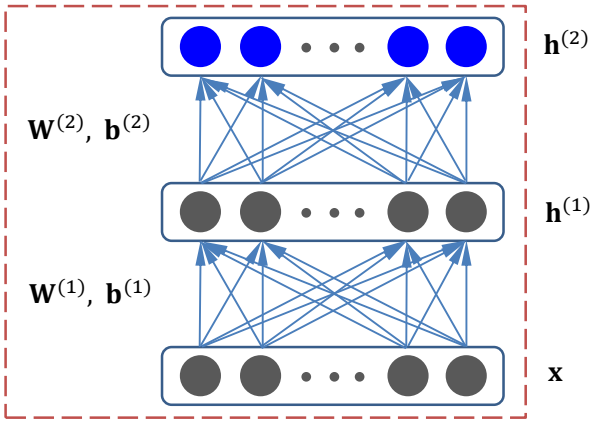

Figure 2. The network architecture used in our method. The input to the network is $\mathbf{x}$, and the output of the hidden layer and the top layer is $\mathbf{h}^{(1)}$ and $\mathbf{h}^{(2)}$, respectively. Here $\mathbf{W}^{(m)}$ and $\mathbf{b}^{(m)}$ are the parameters of the network to be learned, $1 \leq m \leq 2$.

3.2. Deep Metric Learning

Unlike most previous metric learning methods which usually seek a single linear distance to transform sample into a linear feature space, we construct a deep neural network to compute the representations of each sample $\mathbf{x}$ by passing it to multiple layers of nonlinear transformations, as shown in Figure 2. The key advantage of using such a network to map $\mathbf{x}$ is the nonlinear mapping function can be explicitly obtained. Assume there are $M+1$ layers in the designed network and $p^{(m)}$ units in the mth layer, where $m = 1, 2, \cdots, M$. The output of $\mathbf{x}$ at the mth layer is computed as:

$$f^{(m)}(\mathbf{x}) = \mathbf{h}^{(m)} = \varphi\left(\mathbf{W}^{(m)}\mathbf{h}^{(m-1)} + \mathbf{b}^{(m)}\right) \in \mathbb{R}^{p^{(m)}}, \quad (1)$$

where $\mathbf{W}^{(m)} \in \mathbb{R}^{p^{(m)} \times p^{(m-1)}}$ and $\mathbf{b}^{(m)} \in \mathbb{R}^{p^{(m)}}$ are the weight matrix and bias of the parameters in this layer; and φ is a nonlinear activation function which operates component-wisely, such as widely used *tanh* or *sigmoid* functions. The nonlinear mapping $f^{(m)} : \mathbb{R}^d \mapsto \mathbb{R}^{p^{(m)}}$ is a function parameterized by $\{\mathbf{W}^{(i)}\}_{i=1}^m$ and $\{\mathbf{b}^{(i)}\}_{i=1}^m$. For the first layer, we assume $\mathbf{h}^{(0)} = \mathbf{x}$ and $p^{(0)} = d$.

For each pair of samples $\mathbf{x}_i$ and $\mathbf{x}_j$, they can be finally represented as $f^{(m)}(\mathbf{x}_i)$ and $f^{(m)}(\mathbf{x}_j)$ at the mth layer of our designed network, and their distance metric can be measured by computing the squared Euclidean distance between the representations $f^{(m)}(\mathbf{x}_i)$ and $f^{(m)}(\mathbf{x}_j)$ at the mth layer:

$$d^2_{f^{(m)}}(\mathbf{x}_i, \mathbf{x}_j) = \left\| f^{(m)}(\mathbf{x}_i) - f^{(m)}(\mathbf{x}_j) \right\|_2^2. \quad (2)$$

Following the graph embedding framework, we enforce the marginal fisher analysis criterion [35] on the output of all the training samples at the top layer and formulate a strongly-supervised deep metric learning method as:

$$\min_{f^{(M)}} J = S_c^{(M)} - \alpha\, S_b^{(M)} + \gamma \sum_{m=1}^{M} \left(\left\|\mathbf{W}^{(m)}\right\|_F^2 + \left\|\mathbf{b}^{(m)}\right\|_2^2 \right), \quad (3)$$

where α ($\alpha > 0$) is a free parameter which balances the important between intra-class compactness and interclass separability; $\|\mathbf{Z}\|_F$ denotes the Frobenius norm of the matrix $\mathbf{Z}$; γ ($\gamma > 0$) is a tunable positive regularization parameter; $S_c^{(m)}$ and $S_b^{(m)}$ define the intra-class compactness and the interclass separability, which are defined as follows:

$$S_c^{(m)} = \frac{1}{Nk_1} \sum_{i=1}^{N} \sum_{j=1}^{N} P_{ij}\, d^2_{f^{(m)}}(\mathbf{x}_i, \mathbf{x}_j), \quad (4)$$

$$S_b^{(m)} = \frac{1}{Nk_2} \sum_{i=1}^{N} \sum_{j=1}^{N} Q_{ij}\, d^2_{f^{(m)}}(\mathbf{x}_i, \mathbf{x}_j), \quad (5)$$

where P_{ij} is set as one if $\mathbf{x}_j$ is one of k_1-*intra-class* nearest neighbors of $\mathbf{x}_i$, and zero otherwise; and Q_{ij} is set as one if $\mathbf{x}_j$ is one of k_2-*interclass* nearest neighbors of $\mathbf{x}_i$, and zero otherwise.

3.3. Deep Transfer Metric Learning

Given target domain data $\mathcal{X}_t$ and source domain data $\mathcal{X}_s$, their probability distributions are usually different in the original feature space when they are captured from different datasets. To reduce the distribution difference, it is desirable to make the probability distribution of the source domain and that of the target domain be as close as possible in the transformed space. To achieve this, we apply the Maximum Mean Discrepancy (MMD) criterion [13, 27] to measure their distribution difference at the mth layer, which is defined as as follows:

$$D_{ts}^{(m)}(\mathcal{X}_t, \mathcal{X}_s) = \left\| \frac{1}{N_t} \sum_{i=1}^{N_t} f^{(m)}(\mathbf{x}_{ti}) - \frac{1}{N_s} \sum_{i=1}^{N_s} f^{(m)}(\mathbf{x}_{si}) \right\|_2^2. \quad (6)$$

By combining (3) and (6), we formulate DTML as the following optimization problem:

$$\min_{f^{(M)}} J = S_c^{(M)} - \alpha\, S_b^{(M)} + \beta\, D_{ts}^{(M)}(\mathcal{X}_t, \mathcal{X}_s) + \gamma \sum_{m=1}^{M} \left(\left\|\mathbf{W}^{(m)}\right\|_F^2 + \left\|\mathbf{b}^{(m)}\right\|_2^2 \right), \quad (7)$$

where β ($\beta \geq 0$) is a regularization parameter.

To solve the optimization problem in (7), we employ the stochastic sub-gradient descent method to obtain the parameters $\mathbf{W}^{(m)}$ and $\mathbf{b}^{(m)}$. The gradients of the objective function J in (7) with respect to the parameters $\mathbf{W}^{(m)}$ and $\mathbf{b}^{(m)}$ are computed as follows:

$$\begin{aligned}
&\frac{\partial J}{\partial \mathbf{W}^{(m)}} \\
&= \frac{2}{Nk_1}\sum_{i=1}^{N}\sum_{j=1}^{N} P_{ij}\Big(\mathbf{L}_{ij}^{(m)}\mathbf{h}_i^{(m-1)^T} + \mathbf{L}_{ji}^{(m)}\mathbf{h}_j^{(m-1)^T}\Big) \\
&- \frac{2\alpha}{Nk_2}\sum_{i=1}^{N}\sum_{j=1}^{N} Q_{ij}\Big(\mathbf{L}_{ij}^{(m)}\mathbf{h}_i^{(m-1)^T} + \mathbf{L}_{ji}^{(m)}\mathbf{h}_j^{(m-1)^T}\Big) \\
&+ 2\beta\Big(\frac{1}{N_t}\sum_{i=1}^{N_t}\mathbf{L}_{ti}^{(m)}\mathbf{h}_{ti}^{(m-1)^T} + \frac{1}{N_s}\sum_{i=1}^{N_s}\mathbf{L}_{si}^{(m)}\mathbf{h}_{si}^{(m-1)^T}\Big) \\
&+ 2\gamma\,\mathbf{W}^{(m)},
\end{aligned} \tag{8}$$

$$\begin{aligned}
\frac{\partial J}{\partial \mathbf{b}^{(m)}} &= \frac{2}{Nk_1}\sum_{i=1}^{N}\sum_{j=1}^{N} P_{ij}\Big(\mathbf{L}_{ij}^{(m)} + \mathbf{L}_{ji}^{(m)}\Big) \\
&- \frac{2\alpha}{Nk_2}\sum_{i=1}^{N}\sum_{j=1}^{N} Q_{ij}\Big(\mathbf{L}_{ij}^{(m)} + \mathbf{L}_{ji}^{(m)}\Big) \\
&+ 2\beta\Big(\frac{1}{N_t}\sum_{i=1}^{N_t}\mathbf{L}_{ti}^{(m)} + \frac{1}{N_s}\sum_{i=1}^{N_s}\mathbf{L}_{si}^{(m)}\Big) \\
&+ 2\gamma\,\mathbf{b}^{(m)},
\end{aligned} \tag{9}$$

where the updating equations are computed as follows:

$$\begin{aligned}
\mathbf{L}_{ij}^{(M)} &= \Big(\mathbf{h}_i^{(M)} - \mathbf{h}_j^{(M)}\Big) \odot \varphi'\Big(\mathbf{z}_i^{(M)}\Big), \\
\mathbf{L}_{ji}^{(M)} &= \Big(\mathbf{h}_j^{(M)} - \mathbf{h}_i^{(M)}\Big) \odot \varphi'\Big(\mathbf{z}_j^{(M)}\Big), \\
\mathbf{L}_{ij}^{(m)} &= \Big(\mathbf{W}^{(m+1)^T}\mathbf{L}_{ij}^{(m+1)}\Big) \odot \varphi'\Big(\mathbf{z}_i^{(m)}\Big), \\
\mathbf{L}_{ji}^{(m)} &= \Big(\mathbf{W}^{(m+1)^T}\mathbf{L}_{ji}^{(m+1)}\Big) \odot \varphi'\Big(\mathbf{z}_j^{(m)}\Big), \\
\mathbf{L}_{ti}^{(M)} &= \Big(\frac{1}{N_t}\sum_{j=1}^{N_t}\mathbf{h}_{tj}^{(M)} - \frac{1}{N_s}\sum_{j=1}^{N_s}\mathbf{h}_{sj}^{(M)}\Big) \odot \varphi'\Big(\mathbf{z}_{ti}^{(M)}\Big), \\
\mathbf{L}_{si}^{(M)} &= \Big(\frac{1}{N_s}\sum_{j=1}^{N_s}\mathbf{h}_{sj}^{(M)} - \frac{1}{N_t}\sum_{j=1}^{N_t}\mathbf{h}_{tj}^{(M)}\Big) \odot \varphi'\Big(\mathbf{z}_{si}^{(M)}\Big), \\
\mathbf{L}_{ti}^{(m)} &= \Big(\mathbf{W}^{(m+1)^T}\mathbf{L}_{ti}^{(m+1)}\Big) \odot \varphi'\Big(\mathbf{z}_{ti}^{(m)}\Big), \\
\mathbf{L}_{si}^{(m)} &= \Big(\mathbf{W}^{(m+1)^T}\mathbf{L}_{si}^{(m+1)}\Big) \odot \varphi'\Big(\mathbf{z}_{si}^{(m)}\Big),
\end{aligned}$$

where $m = 1, 2, \cdots, M-1$. Here the operation $\odot$ denotes the element-wise multiplication, and $\mathbf{z}_i^{(m)}$ is given as $\mathbf{z}_i^{(m)} = \mathbf{W}^{(m)}\mathbf{h}_i^{(m-1)} + \mathbf{b}^{(m)}$.

Algorithm 1: DTML

Input: Training set: labeled source domain data $\mathcal{X}_s$ and unlabeled target domain data $\mathcal{X}_t$; Parameters: α, β, γ, M, k_1, k_2, learning rate λ, convergence error ε, and total iterative number T.

for $k = 1, 2, \cdots, T$ **do**
- Do forward propagation to all data points;
- Compute compactness $S_c^{(M)}$ by (4);
- Compute separability $S_b^{(M)}$ by (5);
- Obtain MMD term $D_{ts}^{(M)}(\mathcal{X}_t, \mathcal{X}_s)$ by (6);
- **for** $m = M, M-1, \cdots, 1$ **do**
  - Compute $\partial J/\partial \mathbf{W}^{(m)}$ and $\partial J/\partial \mathbf{b}^{(m)}$ by back-propagation using (8) and (9);
- **end**
- *// Updating weights and biases*
- **for** $m = 1, 2, \cdots, M$ **do**
  - $\mathbf{W}^{(m)} \longleftarrow \mathbf{W}^{(m)} - \lambda\ \partial J/\partial \mathbf{W}^{(m)}$;
  - $\mathbf{b}^{(m)} \longleftarrow \mathbf{b}^{(m)} - \lambda\ \partial J/\partial \mathbf{b}^{(m)}$;
- **end**
- $\lambda \longleftarrow 0.95 \times \lambda$; *// Reducing the learning rate*
- Obtain J_k by (7);
- If $|J_k - J_{k-1}| < \varepsilon$, go to **Output**.

end

Output: Weights and biases $\{\mathbf{W}^{(m)}, \mathbf{b}^{(m)}\}_{m=1}^{M}$.

Then, $\mathbf{W}^{(m)}$ and $\mathbf{b}^{(m)}$ can be updated by using the gradient descent algorithm as follows until convergence:

$$\mathbf{W}^{(m)} = \mathbf{W}^{(m)} - \lambda\,\frac{\partial J}{\partial \mathbf{W}^{(m)}}, \tag{10}$$

$$\mathbf{b}^{(m)} = \mathbf{b}^{(m)} - \lambda\,\frac{\partial J}{\partial \mathbf{b}^{(m)}}, \tag{11}$$

where λ is the learning rate.

Algorithm 1 summarizes the detailed optimization procedure of the proposed DTML method.

4. Deeply Supervised Transfer Metric Learning

The objective function of DTML defined in (7) only considers the supervised information of training samples at the top layer of the network, which ignore the discriminative information of the output at the hidden layers. To address this, we further propose a deeply supervised transfer metric learning (DSTML) method to better exploit discriminative information from the output of all layers. We formulate the following optimization problem:

$$\min_{f^{(M)}} J = J^{(M)} + \sum_{m=1}^{M-1}\omega^{(m)}\,h\big(J^{(m)} - \tau^{(m)}\big), \tag{12}$$

where

$$J^{(m)} = S_c^{(m)} - \alpha\, S_b^{(m)} + \beta\, D_{ts}^{(m)}(\mathcal{X}_t, \mathcal{X}_s) + \gamma \left(\left\| \mathbf{W}^{(m)} \right\|_F^2 + \left\| \mathbf{b}^{(m)} \right\|_2^2 \right), \quad (13)$$

is the objective function of DTML applied at the mth layer. Here $J^{(M)}$ is the loss of the top layer and $J^{(m)}$ is the loss of the mth *hidden* layer, $m = 1, 2, \cdots, M-1$. The hinge loss function is used to measure the loss, which is defined as: $h(x) = \max(x, 0)$; $\tau^{(m)}$ is a positive threshold, which controls the loss $J^{(m)}$ to show it plays the role in the learning procedure; and $\omega^{(m)}$ balances the importance of the loss obtained at the top layer and the mth hidden layer. The second term in (12) will disappear during the learning procedure if the overall loss of the mth hidden layer is below the threshold $\tau^{(m)}$.

The gradient of the objective function J in (12) with respect to the parameters $\mathbf{W}^{(m)}$ and $\mathbf{b}^{(m)}$ at the top layer are computed as follows:

$$\frac{\partial J}{\partial \mathbf{W}^{(M)}} = \frac{\partial J^{(M)}}{\partial \mathbf{W}^{(M)}}, \quad (14)$$

$$\frac{\partial J}{\partial \mathbf{b}^{(M)}} = \frac{\partial J^{(M)}}{\partial \mathbf{b}^{(M)}}, \quad (15)$$

For other layers $m = 1, 2, \cdots, M-1$, they are computed as follows:

$$\frac{\partial J}{\partial \mathbf{W}^{(m)}} = \frac{\partial J^{(M)}}{\partial \mathbf{W}^{(m)}} + \sum_{\ell=m}^{M-1} \omega^{(\ell)}\, h'\big(J^{(\ell)} - \tau^{(\ell)}\big)\, \frac{\partial J^{(\ell)}}{\partial \mathbf{W}^{(m)}}, \quad (16)$$

$$\frac{\partial J}{\partial \mathbf{b}^{(m)}} = \frac{\partial J^{(M)}}{\partial \mathbf{b}^{(m)}} + \sum_{\ell=m}^{M-1} \omega^{(\ell)}\, h'\big(J^{(\ell)} - \tau^{(\ell)}\big)\, \frac{\partial J^{(\ell)}}{\partial \mathbf{b}^{(m)}}, \quad (17)$$

where $h'(x)$ is the derivative of $h(x)$, and we set $h'(x) = 0$ for the non-differentiability point $x = 0$.

For $1 \le m \le \ell \le M$, we have

$$\begin{aligned} \frac{\partial J^{(\ell)}}{\partial \mathbf{W}^{(m)}} &= \frac{2}{Nk_1} \sum_{i=1}^{N} \sum_{j=1}^{N} P_{ij} \Big(\mathbf{L}_{ij}^{(m)} \mathbf{h}_i^{(m-1)^T} + \mathbf{L}_{ji}^{(m)} \mathbf{h}_j^{(m-1)^T} \Big) \\ &- \frac{2\alpha}{Nk_2} \sum_{i=1}^{N} \sum_{j=1}^{N} Q_{ij} \Big(\mathbf{L}_{ij}^{(m)} \mathbf{h}_i^{(m-1)^T} + \mathbf{L}_{ji}^{(m)} \mathbf{h}_j^{(m-1)^T} \Big) \\ &+ 2\beta \Big(\frac{1}{N_t} \sum_{i=1}^{N_t} \mathbf{L}_{ti}^{(m)} \mathbf{h}_{ti}^{(m-1)^T} + \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{L}_{si}^{(m)} \mathbf{h}_{si}^{(m-1)^T} \Big) \\ &+ 2\gamma\, \delta(\ell - m)\, \mathbf{W}^{(\ell)}, \end{aligned} \quad (18)$$

$$\begin{aligned} \frac{\partial J^{(\ell)}}{\partial \mathbf{b}^{(m)}} &= \frac{2}{Nk_1} \sum_{i=1}^{N} \sum_{j=1}^{N} P_{ij} \Big(\mathbf{L}_{ij}^{(m)} + \mathbf{L}_{ji}^{(m)} \Big) \\ &- \frac{2\alpha}{Nk_2} \sum_{i=1}^{N} \sum_{j=1}^{N} Q_{ij} \Big(\mathbf{L}_{ij}^{(m)} + \mathbf{L}_{ji}^{(m)} \Big) \\ &+ 2\beta \Big(\frac{1}{N_t} \sum_{i=1}^{N_t} \mathbf{L}_{ti}^{(m)} + \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{L}_{si}^{(m)} \Big) \\ &+ 2\gamma\, \delta(\ell - m) \mathbf{b}^{(\ell)}, \end{aligned} \quad (19)$$

where the delta function $\delta(x) = 0$ holds except $\delta(x) = 1$ at point $x = 0$, and the updating equations for all layers $1 \le m \le \ell - 1$ are computed as follows:

$$\begin{aligned} \mathbf{L}_{ij}^{(\ell)} &= \Big(\mathbf{h}_i^{(\ell)} - \mathbf{h}_j^{(\ell)} \Big) \odot \varphi' \Big(\mathbf{z}_i^{(\ell)} \Big), \\ \mathbf{L}_{ji}^{(\ell)} &= \Big(\mathbf{h}_j^{(\ell)} - \mathbf{h}_i^{(\ell)} \Big) \odot \varphi' \Big(\mathbf{z}_j^{(\ell)} \Big), \\ \mathbf{L}_{ij}^{(m)} &= \Big(\mathbf{W}^{(m+1)^T} \mathbf{L}_{ij}^{(m+1)} \Big) \odot \varphi' \Big(\mathbf{z}_i^{(m)} \Big), \\ \mathbf{L}_{ji}^{(m)} &= \Big(\mathbf{W}^{(m+1)^T} \mathbf{L}_{ji}^{(m+1)} \Big) \odot \varphi' \Big(\mathbf{z}_j^{(m)} \Big), \\ \mathbf{L}_{ti}^{(\ell)} &= \Big(\frac{1}{N_t} \sum_{j=1}^{N_t} \mathbf{h}_{tj}^{(\ell)} - \frac{1}{N_s} \sum_{j=1}^{N_s} \mathbf{h}_{sj}^{(\ell)} \Big) \odot \varphi' \Big(\mathbf{z}_{ti}^{(\ell)} \Big), \\ \mathbf{L}_{si}^{(\ell)} &= \Big(\frac{1}{N_s} \sum_{j=1}^{N_s} \mathbf{h}_{sj}^{(\ell)} - \frac{1}{N_t} \sum_{j=1}^{N_t} \mathbf{h}_{tj}^{(\ell)} \Big) \odot \varphi' \Big(\mathbf{z}_{si}^{(\ell)} \Big), \\ \mathbf{L}_{ti}^{(m)} &= \Big(\mathbf{W}^{(m+1)^T} \mathbf{L}_{ti}^{(m+1)} \Big) \odot \varphi' \Big(\mathbf{z}_{ti}^{(m)} \Big), \\ \mathbf{L}_{si}^{(m)} &= \Big(\mathbf{W}^{(m+1)^T} \mathbf{L}_{si}^{(m+1)} \Big) \odot \varphi' \Big(\mathbf{z}_{si}^{(m)} \Big). \end{aligned}$$

5. Experiments

In this section, we evaluate the DTML and DSTML methods on two visual recognition tasks: cross-dataset face verification and cross-dataset person re-identification. The followings describe the detailed experiments and results.

5.1. Face Verification

Face verification aims to decide whether a given pair of face images are from the same person or not. In this section, we conducted cross-dataset unconstrained face verification where face images were captured in wild conditions so that significant variations such as varying expression, pose, lighting, and background occur the captured images.

Datasets and Experimental Settings: We used the Labeled Faces in the Wild (LFW) [18] and the Wide and Deep Reference (WDRef) [6] datasets for cross-dataset face verification. LFW is a challenging dataset which was developed for studying the problem of unconstrained face verification. There are 13233 face images of 5749 persons in the LFW dataset, which were collected from the internet and 1680 people have two or more images. This dataset was divided

Figure 3. Sampled face images from LFW and WDRef [6].

into 10 non-overlapping folds and the 10-fold cross validation evaluation was used to test the performance of different face verification methods. For each fold, there are 300 matched (positive) pairs and 300 mismatched (negative) image pairs, respectively.

The WDRef [6] dataset contains 99773 face images of 2995 people, and 2065 subjects have at least 15 images. Since face image captured in WDRef are also from the internet, face images from these two datasets are generally similar. There is no overlapped subject between the LFW and WDRef datasets. Figure 3 shows some sample images from these two datasets. In our experiments, we applied a subset which is randomly sampled from WDRef which consists of 15000 images (1500 subjects and each subject has 10 images) to learn the discriminative metric network.

In our settings, we selected the LFW dataset as the target domain data and the WDRef dataset as the source domain data. For each face image in these two datasets, we used the conventional local binary patterns (LBP) feature [1] provided by the authors [6] to represent each face image. Specifically, we first extracted LBP descriptors at five facial landmarks with various scales, and then concatenated them to form a 5900-dimensional feature vector. To further remove the redundancy, each feature vector is reduced to 500 dimension by principal component analysis (PCA) where the projection is learnt on the source domain data. For our proposed DTML and DSTML methods, we designed a deep network with three layers ($M = 2$), and neural nodes from bottom to top layer are set as: $500 \rightarrow 400 \rightarrow 300$. The *tanh* function was used as the nonlinear activation function in our methods, and the parameters α, β, γ, $\omega^{(1)}$, $\tau^{(1)}$, k_1 and k_2 were empirically set as 0.1, 10, 0.1, 1, 0, 5 and 10, respectively. The initialized value of the learning rate λ was set as 0.2, and then it gradually reduced by multiplying a factor 0.95 in each iteration. We initialized $\mathbf{W}^{(m)}$ as a matrix with ones on the main diagonal and zeros elsewhere, and set the bias $\mathbf{b}^{(m)}$ as $\mathbf{0}$ for all $m = 1, 2, \cdots, M$.

Results: We first compared our DTML with the shallow transfer metric learning (STML) method to show the advantage of the deep network. STML is a special case of our DTML model where the designed neural network only has two layers and the linear activation function $\varphi(x) = x$ was used. Table 1 shows the average verification rate with standard error of DTML, DSTML and STML. We see that

Table 1. Mean verification rate with standard error (%) of different transfer metric learning methods on the LFW dataset.

Method	Accuracy (%)
STML	83.60 ± 0.75
DTML	85.58 ± 0.61
DSTML	$\mathbf{87.32 \pm 0.67}$

Table 2. Mean verification rate with standard error (%) of different metric learning methods when knowledge transfer or no knowledge transfer is used on the LFW dataset.

Method	Transfer	Accuracy (%)
DDML [16]	*no*	83.16 ± 0.80
STML	*yes*	83.60 ± 0.75
STML ($\beta = 0$)	*no*	82.57 ± 0.81
DTML	*yes*	85.58 ± 0.61
DTML ($\beta = 0$)	*no*	83.80 ± 0.55
DSTML	*yes*	$\mathbf{87.32 \pm 0.67}$

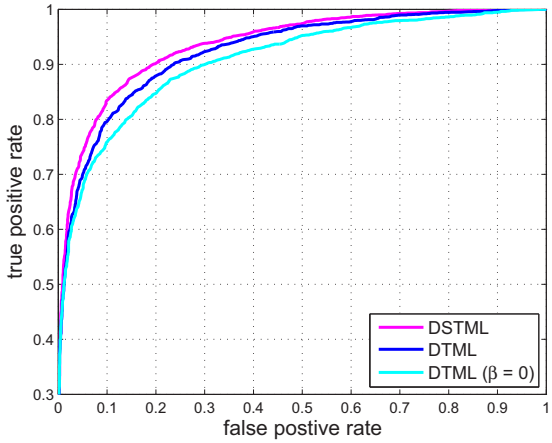

Figure 4. The ROC curves of various methods on LFW dataset.

DTML achieves better performance than STML. The reason is that DTML learns several hierarchical nonlinear transformations while STML learns only one transformation, so that the nonlinearity can be better exploited. Moreover, DSTML improves DTML by 1.7% in terms of the mean verification rate. This is because DSTML utilizes the label information from training sampled in the source domain at both the hidden layers and the top layer while DTML only exploit such information at the top layer, such that more discriminative information is exploited in DSTML.

To show the efficacy of knowledge transfer in DTML, we also compared DTML and STML with their corresponding metric learning methods with no knowledge transfer. Hence, the MMD regularization term in DTML is removed, which means the value of the parameter β is set as 0 in DTML. Table 2 shows the performance comparisons of DTML and shallow TML methods with or with-

Table 3. Top r ranked matching accuracy (%) on the VIPeR dataset with $\#test = 316$ testing persons.

Method	Source	$r = 1$	$r = 5$	$r = 10$	$r = 30$
L_1	-	3.99	8.73	12.59	25.32
L_2	-	4.24	8.92	12.66	25.35
DDML [16]	i-LIDS	5.63	12.91	21.71	41.80
	CAVIAR	5.91	13.53	19.86	37.92
	3DPeS	6.67	17.16	23.87	41.65
DTML ($\beta = 0$)	i-LIDS	5.88	13.72	21.03	41.49
	CAVIAR	6.02	13.81	20.33	38.46
	3DPeS	7.20	18.04	25.96	43.80
DTML	i-LIDS	6.68	15.73	23.20	46.42
	CAVIAR	6.17	13.10	19.65	37.78
	3DPeS	8.51	**19.40**	**27.59**	**47.91**
DSTML	i-LIDS	6.11	16.01	23.51	45.35
	CAVIAR	6.61	16.93	24.40	41.55
	3DPeS	**8.58**	19.02	26.49	46.77

Table 4. Top r ranked matching accuracy (%) on the i-LIDS dataset with $\#test = 60$ testing persons.

Method	Source	$r = 1$	$r = 5$	$r = 10$	$r = 30$
L_1	-	16.51	28.41	38.28	69.32
L_2	-	16.30	28.25	38.40	69.77
DDML [16]	VIPeR	25.32	45.61	60.27	83.31
	CAVIAR	25.67	45.03	61.38	82.56
	3DPeS	28.71	48.55	62.53	83.15
DTML ($\beta = 0$)	VIPeR	26.27	47.59	62.62	85.07
	CAVIAR	26.15	46.87	62.08	84.78
	3DPeS	30.23	51.60	65.21	85.53
DTML	VIPeR	28.90	51.43	65.47	87.23
	CAVIAR	26.23	49.31	63.99	87.76
	3DPeS	31.01	54.51	65.96	88.66
DSTML	VIPeR	28.35	50.81	61.58	84.72
	CAVIAR	28.37	49.68	64.59	88.68
	3DPeS	**33.37**	**54.56**	**68.27**	**89.32**

Table 5. Top r ranked matching accuracy (%) on the CAVIAR dataset with $\#test = 36$ testing persons.

Method	Source	$r = 1$	$r = 5$	$r = 10$	$r = 30$
L_1	-	20.65	36.44	48.52	88.34
L_2	-	20.19	36.43	48.55	87.69
DDML [16]	VIPeR	23.80	42.15	55.61	90.73
	i-LIDS	22.72	41.36	56.92	90.06
	3DPeS	23.85	44.30	57.81	90.27
DTML ($\beta = 0$)	VIPeR	23.71	42.57	56.15	90.55
	i-LIDS	23.09	42.81	58.43	90.41
	3DPeS	25.11	46.71	59.69	91.99
DTML	VIPeR	23.88	42.36	55.60	92.12
	i-LIDS	26.06	47.37	61.70	**94.23**
	3DPeS	26.10	47.80	61.31	93.02
DSTML	VIPeR	26.05	44.33	57.02	92.80
	i-LIDS	25.91	44.47	58.88	93.33
	3DPeS	**28.18**	**49.96**	**63.67**	94.13

Table 6. Top r ranked matching accuracy (%) on the 3DPeS dataset with $\#test = 95$ testing persons.

Method	Source	$r = 1$	$r = 5$	$r = 10$	$r = 30$
L_1	-	26.93	42.21	51.56	69.57
L_2	-	26.95	42.57	51.46	69.84
DDML [16]	VIPeR	29.56	51.03	61.71	78.62
	i-LIDS	27.81	50.29	58.33	77.05
	CAVIAR	30.32	49.36	58.92	79.61
DTML ($\beta = 0$)	VIPeR	30.33	52.18	62.24	82.58
	i-LIDS	29.12	50.07	59.99	78.59
	CAVIAR	31.23	51.88	60.87	81.30
DTML	VIPeR	32.12	**54.36**	**65.92**	84.65
	i-LIDS	32.11	52.08	61.63	79.45
	CAVIAR	31.79	51.92	62.78	81.98
DSTML	VIPeR	32.51	52.97	63.12	83.08
	i-LIDS	31.57	52.54	63.50	84.02
	CAVIAR	**32.53**	54.29	65.28	**84.72**

out knowledge transfer from source domain on the LFW dataset. We see that methods with knowledge transfer consistently improve those methods without knowledge transfer by $1\% \sim 1.7\%$ in terms of the mean verification rate even if the LFW and WDRef datasets are similar. Figure 4 shows the ROC curves of these metric learning methods, we see that transfer metric learning methods outperform those methods without knowledge transfer.

5.2. Person Re-Identification

Person re-identification aims to recognize person across multiple cameras without overlapping views. This task is very challenging because images of the same subject collected in multiple cameras are usually different due to variations of viewpoint, illumination, pose, resolution and occlusion. While many person re-identification methods have been proposed [24, 34, 40], there is no much work on cross-dataset person re-identification. In this subsection, we evaluate our methods on cross dataset person re-identification where only the label information of the source domain is used in the model learning.

Datasets and Experimental Settings: The VIPeR dataset [12] consists of 632 intra-personal image pairs captured outdoor by two different camera views, and most of them contain a viewpoint change of about 90 degrees. The i-LIDS dataset [39] contains 476 images of 119 people captured by five cameras at an airport, and each pedestrian has $2 \sim 8$ images. The CAVIAR dataset [7] contains 1220 images of 72 individuals from two different cameras in an indoor shopping mall, with $10 \sim 20$ images per person as well as large variations in resolutions. The 3DPeS dataset [3] has 1011 images of 192 persons collected from 8 different outdoor cameras with significant changes of viewpoint, and most of individuals appear in three different camera views. Figure 5 shows some example images from these four datasets, respectively.

Figure 5. Example person images from the VIPeR, i-LIDS, CAVIAR and 3DPeS datasets, respectively, where each column shows two images belonging to the same person captured in two different cameras.

In our experiments, all images from these datasets were scaled to 128×48 for feature extraction. For each image, we used two kinds of features descriptor: color and texture histograms. Following the settings in [34], each image was divided into six non-overlapping horizontal stripes. For each stripe, we extracted 16-bins histograms for each color channel of eight channels in the RGB (R, G, B), YUV (Y, U, V) and HSV (H, S) spaces, and computed uniform LBP with 8 neighbors and 16 neighbors respectively. Finally, color and texture histograms from these stripes were concatenated to form a 2580-dimensional feature vector for each image. Then PCA learnt on the source domain data was applied to reduce the dimension of target feature vector into a low dimensional subspace. We also adopted the single-Shot experiment setting [34, 40] to randomly split individuals in each dataset as training set and testing set, and repeated 10 times. For each partition, there were $\#test$ subjects in the testing set, and one image for each person was randomly selected as the gallery set and the remaining images of this person were used as probe images. We also used a network with three layers ($M = 2$), and its neural nodes are given as: $200 \rightarrow 200 \rightarrow 100$ for all datasets. The parameter k_1 and k_2 are set as 3 and 10, respectively[1]. For other parameters, we used the same settings which were used in previous face verification experiments.

Results: Tables 3–6 show cross dataset person re-identification performance of our proposed methods on the VIPeR, i-LIDS, CAVIAR and 3DPeS datasets, respectively. The L_1 and L_2 are two baseline methods which directly use L_1 and L_2 norms to compute distance between a probe image and a gallery image in the target domain. From these tables, we see that our DTML achieve better performance than the metric learning method without knowledge transfer in most of cases, which shows that transfer metric learning can improve the performance of cross dataset person re-identification. We also observe that DSTML obtains the best performance for most cases, which indicates that exploiting discriminative information from more hidden layers rather than the single output layer is more effective to learn good distance metrics.

[1]If the number of image for a given person is less than 3, all intra-class neighbors were used to compute the intra-class variations.

6. Conclusion

In this paper, we have proposed a deep transfer metric learning (DTML) method for cross-dataset visual recognition. By learning a set of hierarchical nonlinear transformations which transfer discriminative knowledge from the labeled source domain to the unlabeled target domain, the proposed method can achieve better performance than existing linear metric learning methods. To better exploit the discriminative information from the source domain, we have further developed a deeply supervised transfer metric learning (DSTML) method by exploiting discriminative information from both the hidden layer and the top output layer to further improve the recognition performance. Experimental results are presented to show the effectiveness of the proposed methods.

Acknowledgements

This work was partly supported by a research grant from the Agency for Science, Technology and Research (A*STAR) of Singapore for the Human Centric Cyber Systems (HCCS) Program at the Advanced Digital Sciences Center (ADSC), and a research grant from the National Research Foundation, Prime Ministers Office, Singapore under its IDM Futures Funding Initiative and administered by the IDM Programme Office at the Rapid-Rich Object Search (ROSE) Lab at the Nanyang Technological University, Singapore.

References

[1] T. Ahonen, S. Member, A. Hadid, M. Pietikanen, and S. Member. Face description with local binary patterns: Application to face recognition. *PAMI*, 28:2037–2041, 2006.

[2] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *JMLR*, 6:1817–1853, 2005.

[3] D. Baltieri, R. Vezzani, and R. Cucchiara. 3dpes: 3d people dataset for surveillance and forensics. In *Proceedings of the 1st International ACM Workshop on Multimedia access to 3D Human Objects*, pages 59–64, 2011.

[4] Y. Bengio. Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[5] X. Cai, C. Wang, B. Xiao, X. Chen, and J. Zhou. Deep nonlinear metric learning with independent subspace analysis for face verification. In *ACM MM*, pages 749–752, 2012.

[6] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun. Bayesian face revisited: A joint formulation. In *ECCV*, pages 566–579, 2012.

[7] D. S. Cheng, M. Cristani, M. Stoppa, L. Bazzani, and V. Murino. Custom pictorial structures for re-identification. In *BMVC*, pages 1–11, 2011.

[8] W. Dai, Q. Yang, G. Xue, and Y. Yu. Boosting for transfer learning. In *ICML*, pages 193–200, 2007.

[9] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *ICML*, pages 209–216, 2007.

[10] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, pages 647–655, 2014.

[11] L. Duan, I. W. Tsang, D. Xu, and S. J. Maybank. Domain transfer SVM for video concept detection. In *CVPR*, pages 1375–1381, 2009.

[12] D. Gray and H. Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *ECCV*, pages 262–275, 2008.

[13] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola. A kernel method for the two-sample-problem. In *NIPS*, pages 513–520, 2006.

[14] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *ICCV*, pages 498–505, 2009.

[15] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[16] J. Hu, J. Lu, and Y. Tan. Discriminative deep metric learning for face verification in the wild. In *CVPR*, pages 1875–1882, 2014.

[17] G. B. Huang, H. Lee, and E. G. Learned-Miller. Learning hierarchical representations for face verification with convolutional deep belief networks. In *CVPR*, pages 2518–2525, 2012.

[18] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[19] M. Köstinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof. Large scale metric learning from equivalence constraints. In *CVPR*, pages 2288–2295, 2012.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[21] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*, pages 3361–3368, 2011.

[22] C. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. In *AISTATS*, pages 562–570, 2015.

[23] J. Lu, J. Hu, X. Zhou, Y. Shang, Y. Tan, and G. Wang. Neighborhood repulsed metric learning for kinship verification. In *CVPR*, pages 2594–2601, 2012.

[24] A. J. Ma, P. C. Yuen, and J. Li. Domain transfer support vector ranking for person re-identification without target camera label information. In *ICCV*, pages 3567–3574, 2013.

[25] H. V. Nguyen and L. Bai. Cosine similarity metric learning for face verification. In *ACCV*, pages 709–720, 2010.

[26] M. Norouzi, D. J. Fleet, and R. Salakhutdinov. Hamming distance metric learning. In *NIPS*, pages 1070–1078, 2012.

[27] S. J. Pan, J. T. Kwok, and Q. Yang. Transfer learning via dimensionality reduction. In *AAAI*, pages 677–682, 2008.

[28] S. J. Pan and Q. Yang. A survey on transfer learning. *TKDE*, 22(10):1345–1359, 2010.

[29] C. Szegedy, A. Toshev, and D. Erhan. Deep neural networks for object detection. In *NIPS*, pages 2553–2561, 2013.

[30] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, pages 1701–1708, 2014.

[31] D. Tran and A. Sorokin. Human activity recognition with metric learning. In *ECCV*, pages 548–561, 2008.

[32] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *JMLR*, 10:207–244, 2009.

[33] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. J. Russell. Distance metric learning with application to clustering with side-information. In *NIPS*, pages 505–512, 2002.

[34] F. Xiong, M. Gou, O. I. Camps, and M. Sznaier. Person re-identification using kernel-based metric learning methods. In *ECCV*, pages 1–16, 2014.

[35] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *PAMI*, 29(1):40–51, 2007.

[36] D. Yeung and H. Chang. A kernel approach for semisupervised metric learning. *TNN*, 18(1):141–149, 2007.

[37] Y. Zhang and D. Yeung. Transfer metric learning by learning task relationships. In *KDD*, pages 1199–1208, 2010.

[38] Y. Zhang and D. Yeung. Transfer metric learning with semi-supervised extension. *TIST*, 3(3):54, 2012.

[39] W. Zheng, S. Gong, and T. Xiang. Associating groups of people. In *BMVC*, pages 1–11, 2009.

[40] W. Zheng, S. Gong, and T. Xiang. Reidentification by relative distance comparison. *PAMI*, 35(3):653–668, 2013.