Optimal Projection Guided Transfer Hashing for Image Retrieval

Ji Liu, Lei Zhang*
Learning Intelligence & Vision Essential Group
School of Microelectronics and Communication Engineering, Chongqing University, China
jiliu@cqu.edu.cn, leizhang@cqu.edu.cn

Abstract

Recently, learning to hash has been widely studied for image retrieval thanks to the computation and storage efficiency of binary codes. For most existing learning to hash methods, sufficient training images are required and used to learn precise hashing codes. However, in some real-world applications, there are not always sufficient training images in the domain of interest. In addition, some existing supervised approaches need a amount of labeled data, which is an expensive process in terms of time, labor and human expertise. To handle such problems, inspired by transfer learning, we propose a simple yet effective unsupervised hashing method named Optimal Projection Guided Transfer Hashing (GTH) where we borrow the images of other different but related domain i.e., source domain to help learn precise hashing codes for the domain of interest i.e., target domain. Besides, we propose to seek for the maximum likelihood estimation (MLE) solution of the hashing functions of target and source domains due to the domain gap. Furthermore, an alternating optimization method is adopted to obtain the two projections of target and source domains such that the domain hashing disparity is reduced gradually. Extensive experiments on various benchmark databases verify that our method outperforms many state-of-the-art learning to hash methods. The implementation details are available at https://github.com/liuji93/GTH.

Introduction

In recent years, learning to hash algorithms have been proposed to handle the large-scale information retrieval problems in machine learning, computer vision, and big data communities (Wang et al. 2017b). The main goal of hashing techniques is to encode documents, images, and videos to a set of compact binary codes that preserve the feature similarity/dissimilarity in Hamming space. As a result, there will be less storage cost and faster computational speed by using binary features.

However, most existing learning to hash methods are faced with two problems. On one hand, most existing learning to hash methods usually require a large amount of data instances to learn a set of binary hashing codes. However, in some real-world applications, for a domain of interest, i.e., the target domain, the data instances may not be sufficient enough to learn a precise hashing model. Some supervised methods need a large number of labeled images to learn hashing codes. It is well-known that it takes a lot of time, labor and human expertise to tag images. On the other hand, they assume that the distributions of training and testing data are similar, which may not hold in many real-world applications such as cross pose and cross camera cases, etc.

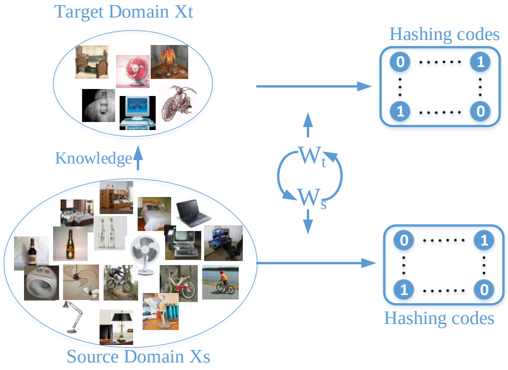


Figure 1: Overview of our GTH. The images from the relevant but different source domain are used to help learn hashing codes for target domain where there are insufficient images that can be used to learn effective hashing codes.

To handle the above problems, inspired by transfer learning, we propose a simple yet effective Optimal Projection Guided Transfer Hashing (GTH) method in this paper. Due to the distribution disparity of source and target domains, we propose to learn two hashing projections for target and source domains respectively in our GTH. Moreover, the knowledge from source domain can be easily used to promote target domain to learn precise hashing codes. In transfer hashing, it is important to guarantee similar images between target and source domains have similar hashing codes. In our GTH, we assume that similar images between target and source domains should mean small discrepancy between hashing projections. To this end, we let the hashing projection (functions) of target domain close to the hashing projection of source domain.

It is easy to adopt minimizing l_2 or l_1 loss between the two hashing projections of source and target domains directly. In other words, in the term of maximum likelihood estimation, we actually assume that errors between two projections of source and target domains obey Gaussian or Laplacian distribution with the l_2 or l_1 loss. However, the data distributions of source and target domains are not similar due to the existence of cross pose, cross camera, and illumination variation, etc. Therefore, the distribution of errors may be far from Gaussian or Laplacian distribution. To improve the above problem, we propose the GTH model from the view of maximum likelihood estimation in this paper. Inspired by (Yang et al. 2011), we design an iteratively weighted l_2 loss for the errors between the projections of source and target domains, which makes our GTH more adaptive to cross-domain case.

Besides, an alternating optimization method is adopted to obtain the two projections of target and source domain such that the domain disparity is reduced gradually. The two different domains can share the hashing projections each other. In other words, the target projection learning is guided by source projection and, in return, the source projection learning is guided by target projection. Finally, the optimal projections of target and source domains will be obtained. The overview of our GTH is shown as Fig. 1. The main contributions and novelties of this paper are summarized as follows.

- Guided by transfer learning, we propose a simple Optimal Projection Guided Transfer Hashing (GTH) method. To the best of our knowledge, there are few methods proposed to handle the problem that there are insufficient training images to learn precise model. We first develop a total unsupervised transfer hashing method to solve cross-domain hashing problem for image retrieval based on conventional machine learning.
- We first propose to learn hashing projections for target and source domains respectively due to the domain disparity. The domain gap is reduced by modeling on hashing projections rather than data level.
- In our GTH, we propose to seek for the maximum likelihood estimation (MLE) solution of the hashing functions of target and source domains due to the domain gap, and design an iteratively weighted l_2 loss for the errors between the projections of source and target domains such that the high error will be punished. Besides, the projections of target and source domain are optimized in a sharing way such that the domain hashing disparity is reduced gradually.
- Extensive experiments on various benchmark databases have been conducted. The experimental results verify that our method outperforms many state-of-the-art learning to hash methods.

Related Work

In this section, we present related works on learning to hash and transfer learning.

Learning to hash

In the past 10 years, various hashing methods have been proposed. Based on whether priori semantic information is used, they can categorized into two major groups: supervised hashing and unsupervised hashing. There are a lot of supervised hashing methods such as LDA hashing (Strecha et al. 2011), Minimal Loss Hashing (Norouzi and Fleet 2011), FastHash (Lin et al. 2014), Kernel-based Supervised Hashing (KSH) (Liu et al. 2012), Supervised Discrete Hashing (SDH) (Shen et al. 2015), the Kernel-based Supervised Discrete Hashing (KSDH) (Shi et al. 2016), and Supervised Quantization for similarity search (SQ) (Wang et al. 2016) that preserve similarity/dissimilarity of intra-class/inter-class images by using semantic information. However, there always lacks label information for model learning due to the high cost of labour and finance in some real-world application situation.

Unsupervised hashing methods aim to explore the intrinsic structure of data to preserve the similarity of neighbors without any supervised information. A number of unsupervised hashing methods have been developed in recent years. Locality-sensitive Hashing (LSH) (Gionis, Indyk, and Motwani 1999), a typical data-independent method, uses a set of randomly generating projection to transform the image features to hashing codes. The representative unsupervised and data-dependent hashing methods include Spectral Hashing (SH) (Weiss, Torralba, and Fergus 2008), Anchor Graph Hashing (AGH) (Liu et al. 2011), Iterative Quantization (ITQ) (Gong et al. 2013), Density Sensitive Hashing (DSH) (Jin et al. 2014), Circulant Binary Embedding (CBE) (Yu et al. 2014), etc. Several ranking-preserved hashing algorithms have been proposed recently to learn more discriminative binary codes e.g., Scalable Graph Hashing (SGH) (Jiang and Li 2015), and Ordinal Constraint Hashing (OCH) (Liu et al. 2018).

Transfer learning

Transfer learning (TL) (Pan and Yang 2010), a new proposed learning conception, aims to transfer knowledge across two different domains such that rich source domain knowledge can be utilized to generate better classifiers on a target domain. In transfer learning, the transferred knowledge can be labels (Zhou et al. 2014), (Yang et al. 2017), features (Zhang and Zhang 2016), (Xu et al. 2017), (Yang et al. 2016), (Wang, Zhang, and Zuo 2017) and cross domain correspondences (Zhang, Zuo, and Zhang 2016), (Wang et al. 2017a). Transfer learning has shown promising results in many machine learning tasks, such as classification and regression. To the best of our knowledge, there are few works on studying transfer learning for hashing. Most of them are based on deep learning (Venkateswara et al. 2017). The recent work (Zhou et al. 2018) proposes a transfer hashing from shallow to deep. Different from their works, we focus on how to transfer knowledge across hashing projection in an unsupervised manner. It is worth noting that the labels in neither of target and source domains are used in our GTH.

Optimal Projection Guided Transfer Hashing

In this section, we present the detailed discussion of Optimal Projection Guided Transfer Hashing (GTH) method.

The objective function of our GTH

Suppose that we have N_t target data points $\mathbf{X}_t = [\mathbf{x}_{t_1}, \mathbf{x}_{t_2}, \cdots, \mathbf{x}_{t_{N_t}}] \in \mathbb{R}^{d\times N_t}$. We aim to learn a set of binary code $\mathbf{B}_t = \{\mathbf{b}_{t_i}\}_{i=1}^{N_t} \in \{-1,1\}^{r\times N_t}$ to well preserve feature information of the original dataset. $\mathbf{b}_{t_i}$ is the corresponding binary codes of $\mathbf{x}_{t_i}$. N_t, d, and r denote the number of the target domain samples, the dimension of each sample, and the code length of binary feature, respectively. Similar with most of learning to hash methods, we also learn hashing projection to map and quantize each $\mathbf{x}_{t_i}$ into a binary codes $\mathbf{b}_{t_i}$. However, when the available target training data is limited, i.e., N_t is small, the binary codes learned by existing learning to hash methods can't perform well. In our GTH, we take advantage of the knowledge (i.e., features) of another known domain (i.e., source domain). Suppose that we have already obtained N_s source data points $\mathbf{X}_s = [\mathbf{x}_{s_1}, \mathbf{x}_{s_2}, \cdots, \mathbf{x}_{s_{N_s}}] \in \mathbb{R}^{d\times N_s}$.

We denote $\mathbf{B}_t = \mathrm{H}(\mathbf{W}_t^{\mathrm{T}}\mathbf{X}_t)$ and $\mathbf{B}_s = \mathrm{H}(\mathbf{W}_s^{\mathrm{T}}\mathbf{X}_s)$ where $\mathbf{W}_t \in \mathbb{R}^{d\times r}$ is hashing projection of target domain and $\mathbf{W}_s \in \mathbb{R}^{d\times r}$ is hashing projection of source domain. $\mathrm{H}(v) = sgn(v)$ equals to 1 if $v \geq 0$ and -1 otherwise. In our GTH, to reduce the distribution discrepancy, we let hashing projection of target domain close to source domain:

$$\min_{\mathbf{W}_t,\mathbf{W}_s} \|\mathbf{W}_t - \mathbf{W}_s\|^2. \tag{1}$$

We denote that $\mathbf{E} = \mathbf{W}_t - \mathbf{W}_s$ represents the error matrix. E_{ij} is one element in the error matrix. As discussed above, from the view of maximum likelihood estimation (MLE), the error matrix follows Gaussian distribution by using the Eq.1. However, the different data distributions of source and target domains may lead to that the probability distribution of error matrix is far from Gaussian distribution. Without loss of generality, we let $\mathbf{e} = [E_{11}, E_{21}, \cdots, E_{d1}, \cdots, E_{1r}, E_{2r}, \cdots, E_{dr}]^{\mathrm{T}}$. Assume that $e_1, e_2, \cdots, e_N$ are independently and identically distributed according to some probability density function (PDF) $f_\theta(e_n)$ where $N = d\times r$ and θ denotes the parameter set that characterizes the distribution. The likelihood estimation can be represented as $L_\theta = \prod_{n=1}^{N} f_\theta(e_n)$ and MLE aims to maximize this likelihood function or minimize the negative log likelihood function: $-\mathrm{lnL}_\theta = \sum_{\mathrm{n}=1}^{\mathrm{N}} \rho_\theta(\mathrm{e_n})$ where $\rho_\theta(e_n) = -\mathrm{lnf}_\theta(\mathrm{e_n})$.

With the above analysis, the Eq. 1 with uncertain probability density function can be transformed into the following minimization problem:

$$\min_{\mathbf{W}_t,\mathbf{W}_s} \sum_{n=1}^{N} \rho_\theta(e_n). \tag{2}$$

In general, we assume that the unknown PDF $f_\theta(e_n)$ is symmetric, and the bigger error will assign a low probability value $f_\theta(e_i) < f_\theta(e_j)$ if $|e_i| > |e_j|$. Therefore, $\rho_\theta(e_n)$ has the following properties: $\rho_\theta(0)$ is the global minimal of $\rho_\theta(e_n)$. Specially, we denote $\rho_\theta(0) = 0$; $\rho_\theta(e_n) = \rho_\theta(-e_n)$; $\rho_\theta(e_i) < \rho_\theta(e_j)$ if $|e_i| < |e_j|$.

Denote that $F_\theta(\mathbf{e}) = \sum_{n=1}^{N} \rho_\theta(e_n)$. We approximate $F_\theta(\mathbf{e})$ by using its first order Taylor expansion in the neighborhood $\mathbf{e}_0$:

$$\widetilde{F_\theta}(\mathbf{e}) = F_\theta(\mathbf{e}_0) + (\mathbf{e}-\mathbf{e}_0)^{\mathrm{T}} F'_\theta(\mathbf{e}_0) + R_1(\mathbf{e}), \tag{3}$$

where $R_1(\mathbf{e})$ is the second-order remained term, and $F'_\theta(\mathbf{e}_0)$ is the derivative of $F_\theta(\mathbf{e}_0)$.

$$R_1(\mathbf{e}) = 0.5(\mathbf{e}-\mathbf{e}_0)^{\mathrm{T}} \mathbf{\Omega} (\mathbf{e}-\mathbf{e}_0). \tag{4}$$

$\mathbf{\Omega}$ is a diagonal matrix and we denote

$$\Omega_{nn} = \rho'_\theta(\Lambda_n)/\Lambda_n = \omega_\theta(\Lambda_n), \tag{5}$$

where we randomly assign a value to Λ_n which satisfies $\Lambda_n \in (0, e_n)$ if $e_n > 0$ otherwise $\Lambda_n \in (e_n, 0)$. $\rho'_\theta(\Lambda_n)$ represents first derivative. Because $\rho_\theta(0)$ is the global minimal of $\rho_\theta(e_n)$, we can get $\rho'_\theta(0) = 0$. We denote $\mathbf{e}_0 = \mathbf{0}$ such that we can obtain the following objective function

$$\widetilde{F_\theta}(\mathbf{e}) = R_1(\mathbf{e}) = 0.5\|\mathbf{\Omega}^{\frac{1}{2}}\mathbf{e}\|^2. \tag{6}$$

It is obvious that each element Ω_{nn} in the diagonal matrix $\mathbf{\Omega}$ can be regarded as a weight coefficient to each error value e_n. We expect that the higher value $|e_n|$ will be assigned a lower weight coefficient Ω_{nn}.

According to (Yang et al. 2011) and (Zhang et al. 2003), we also choose the signmoid function as the weight function

$$\omega_\theta(\Lambda_n) = \exp(\mu\delta - \mu\Lambda_{\mathrm{n}}^2)/(1+\exp(\mu\delta - \mu\Lambda_{\mathrm{n}}^2)), \tag{7}$$

where μ and δ are positive scalars. Parameter μ controls the decreasing rate from 1 to 0, and δ controls the location of demarcation point. For the choice of μ and δ, we just follow (Yang et al. 2011). Considering the Eq.5, Eq.7, and $\rho_\theta(0) = 0$, we obtain $\rho_\theta(\Lambda_n)$ as following

$$\rho_\theta(\Lambda_n) = \frac{-1}{2\mu}(ln(1+\exp(\mu\delta - \mu\Lambda_{\mathrm{n}}^2) - \ln(1+\exp(\mu\delta))). \tag{8}$$

Therefore, we can transform Eq.6 into matrix form as following objective function.

$$\min_{\mathbf{W}_t,\mathbf{W}_s} \frac{1}{2}\|\mathbf{M}^{\frac{1}{2}} \odot (\mathbf{W}_t - \mathbf{W}_s)\|^2. \tag{9}$$

We denote $M_{ij} = \omega_\theta(\widetilde{E_{ij}})$ where we randomly choose a value as $\widetilde{E_{ij}}$ which satisfies $\widetilde{E_{ij}} \in (0, E_{ij})$ if $E_{ij} > 0$ otherwise $\widetilde{E_{ij}} \in (E_{ij}, 0)$. Note that $\mathbf{M}$ is the matrix form of all diagonal elements in $\mathbf{\Omega}$.

It is worth noting that the Eq.9 can be viewed as a inductive model. If we let $\omega_\theta(\widetilde{E_{ij}}) = 2$, the Eq.9 is just Eq.1 which assumes that the errors obey Gaussian distribution. Specially, in this paper, GTH-h refers to Eq.9 with $\omega_\theta(\widetilde{E_{ij}})$ being Eq.7 and GTH-g refers to Eq.9 with $\omega_\theta(\widetilde{E_{ij}}) = 2$.

The quantization loss between hashing codes and its magnitude is used as regularization term in GTH. Besides, we impose orthogonality constraints to hashing projections. The overall objective function is as following

$$\begin{aligned} &\min_{\mathbf{W}_t,\mathbf{W}_s,\mathbf{B}_t,\mathbf{B}_s} \frac{1}{2}\|\mathbf{M}^{\frac{1}{2}} \odot (\mathbf{W}_t - \mathbf{W}_s)\|^2 \\ &+ \frac{\lambda_1}{2}\|\mathbf{B}_t - \mathrm{H}({\mathbf{W}_t}^{\mathrm{T}}\mathbf{X}_t)\|^2 + \frac{\lambda_2}{2}\|\mathbf{B}_s - \mathrm{H}({\mathbf{W}_s}^{\mathrm{T}}\mathbf{X}_s)\|^2 \\ &s.t. \quad {\mathbf{W}_t}^{\mathrm{T}}\mathbf{W}_t = \mathbf{I}, {\mathbf{W}_s}^{\mathrm{T}}\mathbf{W}_s = \mathbf{I}, \end{aligned} \tag{10}$$

where $\lambda_1$and λ_2 denote the regularization coefficients.

Optimization

In this paper, we propose a weighted l_2 loss for the errors between the projections of source and target domains, and update the weight coefficients by using the errors from the last iteration. As the non-convex $sgn(\cdot)$ function makes Eq. 10 a NP-hard problem, we relax the $sgn(x)$ function as its signed magnitude x (Lazebnik 2011). Therefore, the Eq. 10 can be rewritten as

$$\begin{aligned} \min_{\mathbf{W}_t,\mathbf{W}_s,\mathbf{B}_t,\mathbf{B}_s} & \frac{1}{2}\|\mathbf{M}^{\frac{1}{2}} \odot (\mathbf{W}_t - \mathbf{W}_s)\|^2 \\ + \frac{\lambda_1}{2}\|\mathbf{B}_t - & {\mathbf{W}_t}^{\mathrm{T}}\mathbf{X}_t\|^2 + \frac{\lambda_2}{2}\|\mathbf{B}_s - {\mathbf{W}_s}^{\mathrm{T}}\mathbf{X}_s\|^2 \\ s.t. \quad & {\mathbf{W}_t}^{\mathrm{T}}\mathbf{W}_t = \mathbf{I}, {\mathbf{W}_s}^{\mathrm{T}}\mathbf{W}_s = \mathbf{I}. \end{aligned} \tag{11}$$

As mentioned above, we will adopt a relax way to solve problem (10). The solutions for optimization problem (11) can be calculated by alternatingly updating the variables, $\mathbf{W}_t$, $\mathbf{W}_s$, $\mathbf{B}_t$, $\mathbf{B}_s$, and $\mathbf{M}$.

$\mathbf{W}_t$-Step. By fixing $\mathbf{W}_s$, $\mathbf{B}_t$, $\mathbf{B}_s$, and $\mathbf{M}$, the projection of target domain $\mathbf{W}_t$ can be obtained by solving the following subproblem

$$\begin{aligned} \min_{\mathbf{W}_t} \|\mathbf{M}^{\frac{1}{2}} \odot (\mathbf{W}_t - \mathbf{W}_s)\|^2 + \lambda_1\|\mathbf{B}_t - {\mathbf{W}_t}^{\mathrm{T}}\mathbf{X}_t\|^2 \\ s.t. \quad {\mathbf{W}_t}^{\mathrm{T}}\mathbf{W}_t = \mathbf{I}. \end{aligned} \tag{12}$$

Updating $\mathbf{W}_t$ is a typical optimization problem with orthogonality constraints. We apply the optimization procedure in (Wen and Yin 2013) to update $\mathbf{W}_t$. Let $\mathbf{G}_t$ be the partial derivative of the objective function with respect to $\mathbf{W}_t$. $\mathbf{G}_t$ is represented as

$$\mathbf{G}_t = \mathbf{M} \odot (\mathbf{W}_t - \mathbf{W}_s) + \lambda_1(\mathbf{X}_t\mathbf{X}_t^{\mathrm{T}}\mathbf{W}_t - \mathbf{X}_t\mathbf{B}_t^{\mathrm{T}}). \tag{13}$$

To preserve the orthogonality constraint on $\mathbf{W}_t$, we first define the skew-symmetric matrix $\mathbf{Q}_t$ (Armstrong 2005) as $\mathbf{Q}_t = \mathbf{W}_t^{\mathrm{T}}\mathbf{G}_t - \mathbf{G}_t^{\mathrm{T}}\mathbf{W}_t$. Then, we adopt Crank Nicolson like scheme (Wen and Yin 2013) to update the orthogonal matrix $\mathbf{W}_t$:

$$\mathbf{W}_t^{(k+1)} = \mathbf{W}_t^{(k)} - \frac{\tau}{2}(\mathbf{W}_t^{(k+1)} + \mathbf{W}_t^{(k)})\mathbf{Q}_t, \tag{14}$$

where τ denotes the step size. We empirically set $\tau = 0.1$. By solving Eq. 14, we can get

$$\mathbf{W}_t^{(k+1)} = \mathbf{W}_t^{(k)}\mathbf{Q}_t, \tag{15}$$

and $\mathbf{Q}_t^{(k+1)} = (\mathbf{I}+\frac{\tau}{2}\mathbf{Q}_t)^{-1}(\mathbf{I}-\frac{\tau}{2}\mathbf{Q}_t)$. We iteratively update $\mathbf{W}_t$ several times based on Eq. 15 with the Barzilai-Borwein (BB) method (Wen and Yin 2013).

$\mathbf{W}_s$-Step. By fixing $\mathbf{W}_t$, $\mathbf{B}_t$, $\mathbf{B}_s$, and $\mathbf{M}$, the projection of source domain $\mathbf{W}_s$ can be solved as:

$$\begin{aligned} \min_{\mathbf{W}_s} \|\mathbf{M}^{\frac{1}{2}} \odot (\mathbf{W}_t - \mathbf{W}_s)\|^2 + \lambda_2\|\mathbf{B}_s - {\mathbf{W}_s}^{\mathrm{T}}\mathbf{X}_s\|^2 \\ s.t. \quad {\mathbf{W}_s}^{\mathrm{T}}\mathbf{W}_s = \mathbf{I}. \end{aligned} \tag{16}$$

Updating $\mathbf{W}_s$ is the same as $\mathbf{W}_t$. Let $\mathbf{G}_s$ be the partial derivative of the objective function with respect to $\mathbf{W}_s$. $\mathbf{G}_s$ is represented as

$$\mathbf{G}_s = \mathbf{M} \odot (\mathbf{W}_s - \mathbf{W}_t) + \lambda_2(\mathbf{X}_s\mathbf{X}_s^{\mathrm{T}}\mathbf{W}_s - \mathbf{X}_s\mathbf{B}_s^{\mathrm{T}}). \tag{17}$$

Algorithm 1 Optimal Projection Guided Transfer Hashing

```
Input: Target samples $\mathbf{X}_t$ and source samples $\mathbf{X}_s$ parameters $\lambda_1 = 0.1$, and $\lambda_2 = 1$, identity matrix $\mathbf{I}$
Output: $\mathbf{W}_t$, $\mathbf{B}_t$, $\mathbf{W}_s$, and $\mathbf{B}_s$
Initialize: Initialize $\mathbf{W}_t^{(0)}$ and $\mathbf{W}_s^{(0)}$ as the top $r$ eigenvectors of $\mathbf{X}_t\mathbf{X}_t^{\mathrm{T}}$ and $\mathbf{X}_s\mathbf{X}_s^{\mathrm{T}}$ corresponding to the $r$ largest eigenvalues, respectively. $\mathbf{B}_t^{(0)}$ and $\mathbf{B}_s^{(0)}$ are random matrices. $k = 1$.
repeat
    update $\mathbf{M}^{(k)}$: by solving $\omega_\theta(\mathbf{W}_t^{(k-1)} - \mathbf{W}_s^{(k-1)})$;
    update $\mathbf{W}_t^{(k)}$: by solving Eq. 15;
    update $\mathbf{W}_s^{(k)}$: by solving Eq. 19;
    update $\mathbf{B}_t^{(k)}$: by solving Eq. 20;
    update $\mathbf{B}_s^{(k)}$: by solving Eq. 21;
    k=k+1;
until max iterations
```

To preserve the orthogonality constraint on $\mathbf{W}_s$, we define the skew-symmetric matrix $\mathbf{Q}_s$ as $\mathbf{Q}_s = \mathbf{W}_s^{\mathrm{T}}\mathbf{G}_s - \mathbf{G}_s^{\mathrm{T}}\mathbf{W}_s$. Then, we adopt Crank Nicolson like scheme to update the orthogonal matrix $\mathbf{W}_s$:

$$\mathbf{W}_s^{(k+1)} = \mathbf{W}_s^{(k)} - \frac{\tau}{2}(\mathbf{W}_s^{(k+1)} + \mathbf{W}_s^{(k)})\mathbf{Q}_s, \tag{18}$$

where τ denotes the step size. We empirically set $\tau = 0.1$ same as updating $\mathbf{W}_t$. By solving Eq. 18, we can get

$$\mathbf{W}_s^{(k+1)} = \mathbf{W}_s^{(k)}\mathbf{Q}_s, \tag{19}$$

and $\mathbf{Q}_s^{(k+1)} = (\mathbf{I}+\frac{\tau}{2}\mathbf{Q}_s)^{-1}(\mathbf{I}-\frac{\tau}{2}\mathbf{Q}_s)$. We iteratively update $\mathbf{W}_s$ several times based on Eq. 19 with the Barzilai-Borwein (BB) method.

$\mathbf{B}_t$-Step and $\mathbf{B}_s$-Step . As $\mathbf{B}_t$ and $\mathbf{B}_s$ are two binary matrixes, the solutions can be directly obtained as:

$$\mathbf{B}_t = sgn(\mathbf{W}_t^{\mathrm{T}}\mathbf{X}_t). \tag{20}$$

$$\mathbf{B}_s = sgn(\mathbf{W}_s^{\mathrm{T}}\mathbf{X}_s). \tag{21}$$

M-Step. The weight matrix $\mathbf{M}$ is directly computed as following:

$$\mathbf{M} = \omega_\theta(\mathbf{W}_t - \mathbf{W}_s). \tag{22}$$

The overall solving procedures are summarized in Algorithm 1.

Experiment

In this section, extensive experiments are conducted to evaluate the proposed hashing method on image retrieval performance. We perform the experiments on three groups benchmark datasets: PIE-C29&PIE-C05 from **PIE** (Sim, Baker, and Bsat 2002), Amazon&Dslr from **Office** (Saenko et al. 2010), and VOC2007&Caltech101 from **VLCS** (Torralba and Efros 2011). We also choose five state-of-the-art learning-to-hash methods, LSH (Gionis, Indyk, and Motwani 1999), ITQ (Gong et al. 2013), CBE (Yu et al. 2014), DSH (Jin et al. 2014), and OCH (Liu et al. 2018) as baselines. For fair comparison, we introduce a NoDA method acted as OCH method.

Table 1: The MAP scores (%) on PIE, Amazon&Dslr, and VOC2007&Caltech101 databases with varying code length from 16 to 64.

	PIE-C29&PIE-C05					Amazon&Dslr					VOC2007&Caltech101				
Bit	16	24	32	48	64	16	24	32	48	64	16	24	32	48	64
LSH	18.23	21.79	25.26	29.91	32.96	19.69	28.92	35.12	46.72	53.07	11.06	16.51	20.61	27.41	33.12
ITQ	18.17	21.63	23.74	26.82	28.86	43.15	51.74	56.80	62.47	65.84	21.69	28.52	33.46	39.50	42.34
CBE	16.31	22.13	27.10	30.06	32.51	20.82	27.60	36.21	47.52	51.96	11.04	15.64	20.68	26.97	33.84
DSH	17.05	19.60	22.01	25.65	28.12	26.51	32.34	37.39	48.29	50.12	8.69	6.23	13.40	15.56	20.21
OCH	20.75	26.29	28.96	33.33	34.39	41.77	52.41	56.00	62.38	65.45	**32.94**	35.45	38.00	41.46	42.25
NoDA	21.06	24.76	26.51	32.11	32.34	41.64	51.96	57.21	63.29	65.63	30.77	34.81	36.95	40.78	41.80
GTH-g	24.16	28.40	31.69	34.95	35.70	44.16	**53.57**	**57.59**	**63.91**	**66.96**	28.62	**41.20**	46.42	56.59	63.10
GTH-h	**25.45**	**29.42**	**31.76**	**35.25**	**36.56**	**45.23**	52.36	57.26	63.17	65.63	30.05	39.70	**48.14**	**57.33**	**63.53**

Datasets, Settings, and Retrieval evaluation

Description of Datasets: The **PIE** dataset consists of 41,368 face images from 68 subjects as a whole. The images are under five near frontal poses (C05, C07, C09, C27 and C29). We use two subsets chosen from poses C05 and C29. Each image is resized to 32 × 32 and represented by a 1024-dim vector. We use pose C29 (containing 1632 images) as target domain and pose C05 (containing 3332 images) as source domain. Specially, for target domain, we randomly select 500 samples as testing images and the rest samples as training images.

The **Office** dataset is a most popular benchmark dataset for object recognition in the domain adaptation computer vision community. The dataset consists of daily objects in an office environment. **Office** has 3 domains: Amazon (A), Dslr (D), and Webcam (W). We use Amazon with 2817 images as the source domain and Dslr with 498 images as target domain. 100 images from target domain are randomly selected as testing images and the rest images are used as training images. Each image is represented by a 4096-d CNN feature vector (Donahue et al. 2013).

The **VLCS** aggregates photos from Caltech, LabelMe, Pascal VOC 2007 and SUN09. It provides a 5-way multiclass benchmark on the five common classes: ‘bird’, ‘car’, ‘chair’, ‘dog’ and ‘person’. The VOC 2007 dataset containing 3376 images is used as source domain and Caltech containing 1415 images is used as target domain. 100 images from target domain are randomly selected as testing images and the rest images are used as training images. Each image is represented by a 4096-d CNN feature vector (Donahue et al. 2013).

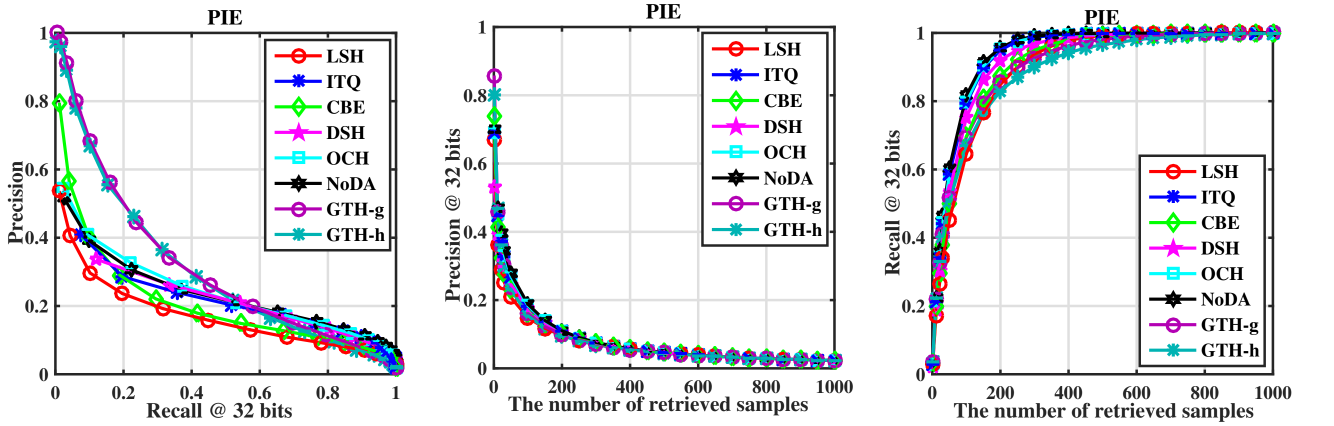
Parameter settings and Implementation details: There are two trade-off parameters in the objective function (10). λ_1 and λ_2 are used to penalize the loss between the binary codes and its signed magnitude. For our GTH, we empirically set λ_1 to 0.1 and λ_2 to 1.

The compared baseline methods are proposed under no domain adaption assumption. For a fair contrast, we use all the source domain data and target domain training data (except the queries on the target domain) as the model input for all compared methods. Besides, we use OCH as a NoDA method. In training phase, we use the training images in target domain as the input of NoDA method. We only focus the retrieval performance on target domain.

Retrieval evaluation: In the Table. 1, we report the MAP scores of all the compared methods and our GTH on PIE-C29&PIE-C05, Amazon&Dslr, and VOC2007&Caltech101 databases. The code lengths are varying from 16 to 64. From the table, we can see that our GTH outperforms compared methods on all databases in most cases. More detailedly, our GTH-h outperforms best compared method NoDA over 4% on PIE-C29&PIE-C05 datasets when the code length is set as 16 bit. On Amazon&Dslr datasets, our GTH-h outperforms best compared method OCH almost 4% with code length set to 16. On the VOC2007&Caltech101 databases, our GTH outperforms much more than the best compared method when the code length is set as 24, 32, 48, and 64. The above results demonstrate the effectiveness of our GTH model and our GTH is more suitable to the condition that there are not enough training images used to learn precise hashing codes in the domain of interest. We also show the PR-curve, Precision and Recall for PIE-C29&PIE-C05 datasets as shown in Fig. 2, Amazon&Dslr dataset as shown in Fig. 3, and VOC2007&Caltech101 databases as shown in Fig. 4. The code length is set to 32 in Figures 2, 3, and 4. From the figures, we can see that our GTH always presents competitive retrieval performance compared to baselines, which demonstrates the efficiency of our GTH.

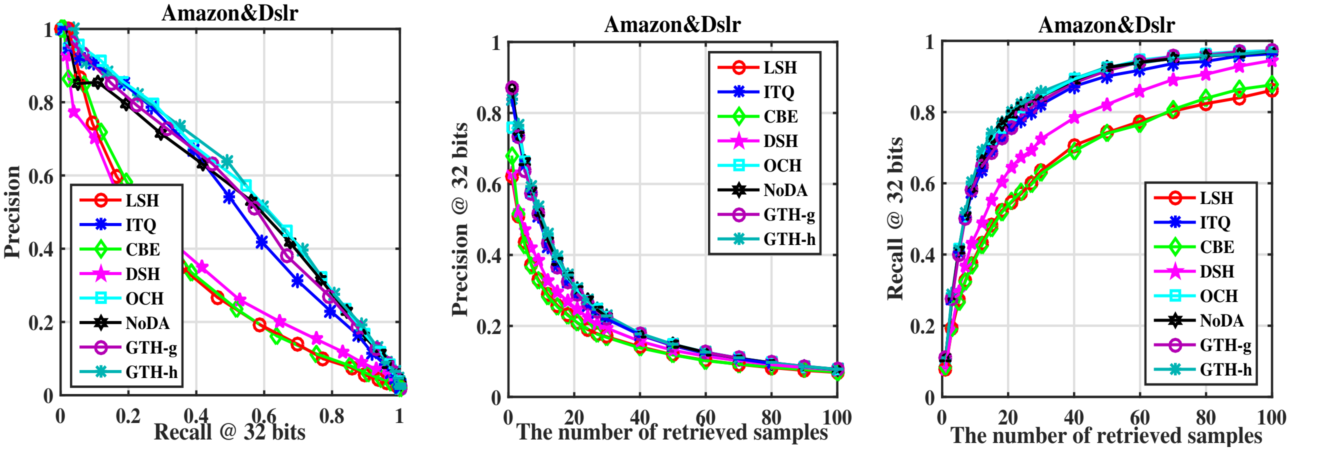
Retrieval evaluation on varying target training numbers

In order to further demonstrate the efficiency of our GTH by using less target training data, we use different numbers of training data on target domain to learn the hashing functions. Specially, we choose 10%, 30%, 50%, and 70% images from training data of target domain as training data i.e., model input. After training, we also use testing hashing codes to search the most similar hashing codes in the whole training samples. The experiments are conducted on PIE-C29&PIE-C05, Amazon&Dslr, and VOC2007&Caltech101 databases respectively. The MAP scores of all compared methods and our GTH are shown in Fig. 5. Due to the input number limitation of OCH method, there are empty MAP scores in some cases. The code length is set as 32. It is worth noting that our GTH always outperforms all the compared methods, which further demonstrates the efficiency of our GTH on the condition that there are less target domain samples to learn precise hashing codes on the domain of interest.



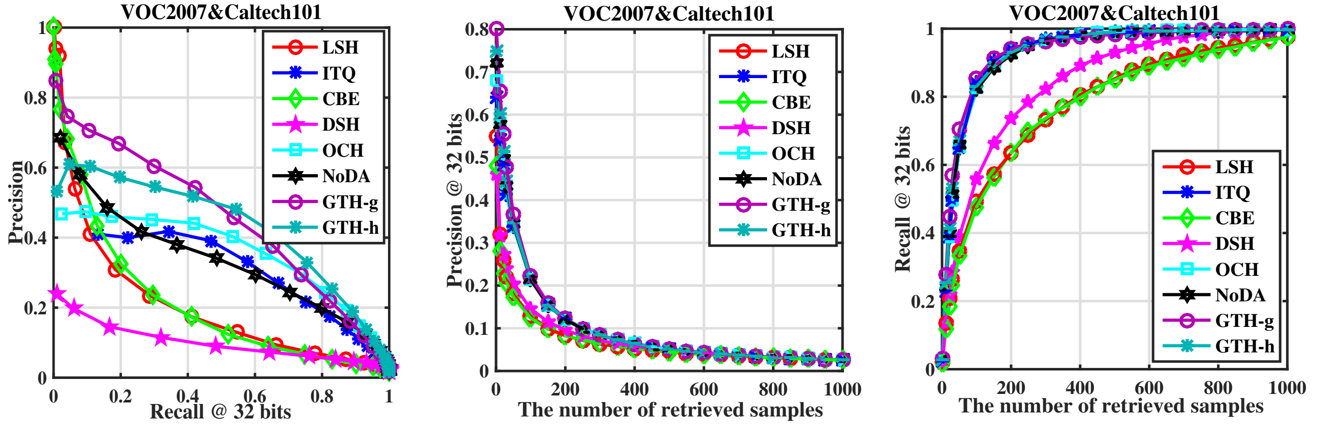
(a) (b) (c)

Figure 2: Retrieval performance on PIE-C29&PIE-C05 datasets @32 bit. (a) Precision and Recall curve; (b) Precision; (c) Recall.



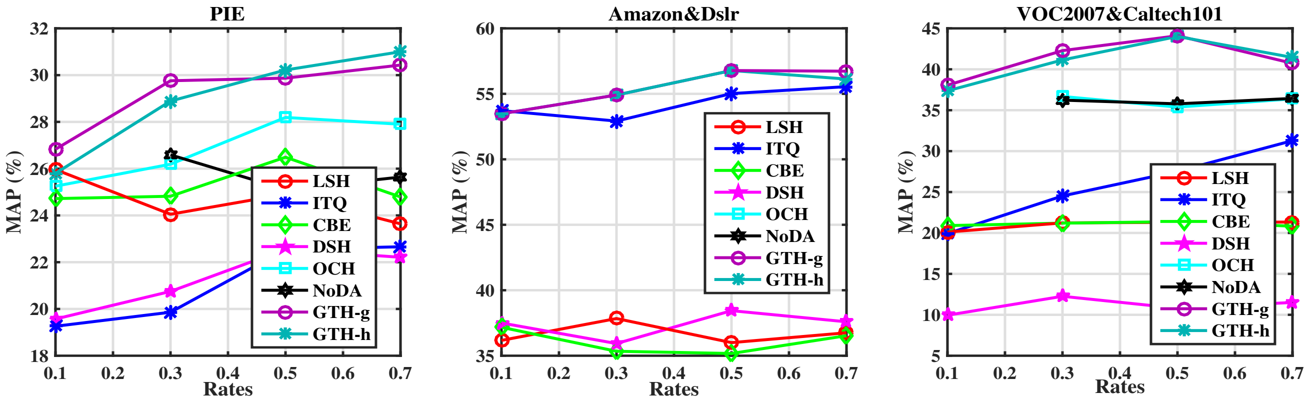
(a) (b) (c)

Figure 3: Retrieval performance on Amazon&Dslr datasets @32 bit. (a) Precision and Recall curve; (b) Precision; (c) Recall.



(a) (b) (c)

Figure 4: Retrieval performance on VOC2007&Caltech101 datasets @32 bit. (a) Precision and Recall curve; (b) Precision; (c) Recall.

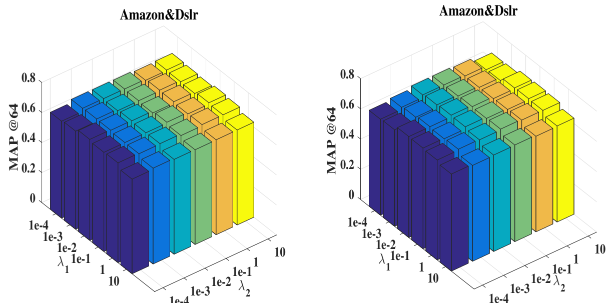


(a)

(b)

(c)

Figure 5: MAP scores @32 bit with varying number training images of target domain. (a) PIE-C29&PIE-C05; (b) Amazon&Dslr; (c) VOC2007&Caltech101.



(a)

(b)

Figure 6: Parameters Sensitivity. (a) GTH-g; (b) GTH-h.

Parameters Sensitivity

In order to further investigate the properties of the proposed method, the retrieval performances versus the different values of regularization parameters, λ_1 and λ_2, are explicitly explored. To clearly show the results, we perform experiments on Amazon&Dslr databases to verify the parameters sensitivity. Specifically, we tune the value of both parameters from $\{0.0001, 0.001, 0.01, 0.1, 1, 10\}$. The MAP scores with code length set to 64 are shown in Fig. 6. We can observe that the performances of our GTH-g and GTH-h models are not very sensitive to the settings of λ_1 and λ_2. Apparently, when the parameters are not very large, the MAP scores of our methods are not severely influenced. This also demonstrates that both regularization terms are indispensable for superior performances. Overall, the proposed models are not sensitive to the parameters in a reasonable range.

Conclusion

We propose a simple but effective transfer hashing method named Optimal Projection Guided Transfer Hashing (GTH) in this paper. Inspired by transfer learning, we propose to borrow the knowledge from a related but different domain. We assume that similar images between target and source domains should mean small discrepancy between hashing projections. Therefore, we let the projections of target and source domain close to each other so that the similar instances between those two domain will be transformed into similar hashing codes. We propose the GTH model from the view of maximum likelihood estimation in this paper and design a iteratively weighted l_2 loss for the errors between the projections of source and target domains, which makes our GTH more adaptive to cross-domain case. Extensive experiments on three groups benchmark databases have been conducted. The experimental results show that our GTH always show much higher retrieval performance when there are much less target samples, which verify that our method outperforms many state-of-the-art learning to hash methods.

Acknowledgement

This work is supported by National Natural Science Fund of China (Grant 61771079) and the Fundamental Research Funds for the Central Universities.

References

Armstrong, A. H. 2005. Numerical solution of partial differential equations. by smith g. d. . pp. viii, 179. 25s. 1965. (oxford university press). *Mathematical Gazette* 50(374):179–449.

Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; and Darrell, T. 2013. Decaf: a deep convolutional activation feature for generic visual recognition. 50(1):I–647.

Gionis, A.; Indyk, P.; and Motwani, R. 1999. Similarity search in high dimensions via hashing. In *International Conference on Very Large Data Bases*, 518–529.

Gong, Y.; Lazebnik, S.; Gordo, A.; and Perronnin, F. 2013. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE*

Transactions on Pattern Analysis and Machine Intelligence 35(12):2916–2929.

Jiang, Q. Y., and Li, W. J. 2015. Scalable graph hashing with feature transformation. In *International Conference on Artificial Intelligence*, 2248–2254.

Jin, Z.; Li, C.; Lin, Y.; and Cai, D. 2014. Density sensitive hashing. *IEEE transactions on cybernetics* 44(8):1362–1371.

Lazebnik, S. 2011. Iterative quantization: A procrustean approach to learning binary codes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 817–824.

Lin, G.; Shen, C.; Shi, Q.; Hengel, A. V. D.; and Suter, D. 2014. Fast supervised hashing with decision trees for high-dimensional data. In *Computer Vision and Pattern Recognition*, 1971–1978.

Liu, W.; Wang, J.; Kumar, S.; and Chang, S. F. 2011. Hashing with graphs. In *International Conference on Machine Learning, ICML 2011, Bellevue, Washington, Usa, June 28 - July*, 1–8.

Liu, W.; Wang, J.; Ji, R.; Jiang, Y. G.; and Chang, S. F. 2012. Supervised hashing with kernels. 2074–2081.

Liu, H.; Ji, R.; Wang, J.; and Shen, C. 2018. Ordinal constraint binary coding for approximate nearest neighbor search. *IEEE Transactions on Pattern Analysis & Machine Intelligence* PP(99):1–1.

Norouzi, M., and Fleet, D. J. 2011. Minimal loss hashing for compact binary codes. In *International Conference on International Conference on Machine Learning*, 353–360.

Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22(10):1345–1359.

Saenko, K.; Kulis, B.; Fritz, M.; and Darrell, T. 2010. Adapting visual category models to new domains. *Eccv Heraklion Greece September*.

Shen, F.; Shen, C.; Liu, W.; and Shen, H. T. 2015. Supervised discrete hashing. In *CVPR*, volume 2, 5.

Shi, X.; Xing, F.; Cai, J.; Zhang, Z.; Xie, Y.; and Yang, L. 2016. *Kernel-Based Supervised Discrete Hashing for Image Retrieval*. In European Conference on Computer Vision.

Sim, T.; Baker, S.; and Bsat, M. 2002. The cmu pose, illumination, and expression (pie) database. In *IEEE International Conference on Automatic Face and Gesture Recognition, 2002. Proceedings*, 46–51.

Strecha, C.; Bronstein, A.; Bronstein, M.; and Fua, P. 2011. Ldahash: Improved matching with smaller descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34(1):66–78.

Torralba, A., and Efros, A. A. 2011. Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition*, 1521–1528.

Venkateswara, H.; Eusebio, J.; Chakraborty, S.; and Panchanathan, S. 2017. Deep hashing network for unsupervised domain adaptation. 5385–5394.

Wang, X.; Zhang, T.; Qi, G. J.; Tang, J.; and Wang, J. 2016. Supervised quantization for similarity search. In *Computer Vision and Pattern Recognition*, 2018–2026.

Wang, B.; Yang, Y.; Xu, X.; Hanjalic, A.; and Shen, H. T. 2017a. Adversarial cross-modal retrieval. In *ACM on Multimedia Conference*, 154–162.

Wang, J.; Zhang, T.; Song, J.; Sebe, N.; and Shen, H. T. 2017b. A survey on learning to hash. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP(99):1–1.

Wang, S.; Zhang, L.; and Zuo, W. 2017. Class-specific reconstruction transfer learning via sparse low-rank constraint. In *ICCVW*, 949–957.

Weiss, Y.; Torralba, A.; and Fergus, R. 2008. Spectral hashing. In *International Conference on Neural Information Processing Systems*, 1753–1760.

Wen, Z., and Yin, W. 2013. A feasible method for optimization with orthogonality constraints. *Mathematical Programming* 142(1-2):397–434.

Xu, Y.; Yang, Y.; Shen, F.; Xu, X.; Zhou, Y.; and Shen, H. T. 2017. Attribute hashing for zero-shot image retrieval. In *IEEE International Conference on Multimedia and Expo*, 133–138.

Yang, M.; Zhang, L.; Yang, J.; and Zhang, D. 2011. Robust sparse coding for face recognition. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 625–632. IEEE.

Yang, Y.; Luo, Y.; Chen, W.; Shen, F.; Shao, J.; and Shen, H. T. 2016. Zero-shot hashing via transferring supervised knowledge. In *ACM on Multimedia Conference*, 1286–1295.

Yang, X.; Wang, M.; Hong, R.; Tian, Q.; and Rui, Y. 2017. Enhancing person re-identification in a self-trained subspace. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 13(3):27.

Yu, F.; Kumar, S.; Gong, Y.; and Chang, S.-F. 2014. Circulant binary embedding. In *International conference on machine learning*, 946–954.

Zhang, L., and Zhang, D. 2016. Robust visual knowledge transfer via extreme learning machine based domain adaptation. *IEEE Transactions on Image Processing* 25(10):4959–4973.

Zhang, J.; Jin, R.; Yang, Y.; and Hauptmann, A. G. 2003. Modified logistic regression: an approximation to svm and its applications in large-scale text categorization. In *Twentieth International Conference on International Conference on Machine Learning*, 888–895.

Zhang, L.; Zuo, W.; and Zhang, D. 2016. Lsdt: Latent sparse domain transfer learning for visual adaptation. *IEEE Trans Image Process* 25(3):1177–1191.

Zhou, J. T.; Pan, S. J.; Tsang, I. W.; and Yan, Y. 2014. Hybrid heterogeneous transfer learning through deep learning. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2213–2219.

Zhou, J. T.; Zhao, H.; Peng, X.; Fang, M.; Qin, Z.; and Goh, R. S. M. 2018. Transfer hashing: From shallow to deep. *IEEE Transactions on Neural Networks and Learning Systems*.