Probability Weighted Compact Feature for Domain Adaptive Retrieval

Fuxiang Huang[1], Lei Zhang[1(✉)], Yang Yang[2], Xichuan Zhou[1]

[1]Learning Intelligence & Vision Essential (LiVE) Group

[1]School of Microelectronics and Communication Engineering, Chongqing University, China

[2]University of Electronic Science and Technology of China

{huangfuxiang, leizhang}@cqu.edu.cn, dlyyang@gmail.com, zxc@cqu.edu.cn

Abstract

Domain adaptive image retrieval includes single-domain retrieval and cross-domain retrieval. Most of the existing image retrieval methods only focus on single-domain retrieval, which assumes that the distributions of retrieval databases and queries are similar. However, in practical application, the discrepancies between retrieval databases often taken in ideal illumination/pose/background/camera conditions and queries usually obtained in uncontrolled conditions are very large. In this paper, considering the practical application, we focus on challenging cross-domain retrieval. To address the problem, we propose an effective method named Probability Weighted Compact Feature Learning (PWCF), which provides inter-domain correlation guidance to promote cross-domain retrieval accuracy and learns a series of compact binary codes to improve the retrieval speed. First, we derive our loss function through the Maximum A Posteriori Estimation (MAP): Bayesian Perspective (BP) induced focal-triplet loss, BP induced quantization loss and BP induced classification loss. Second, we propose a common manifold structure between domains to explore the potential correlation across domains. Considering the original feature representation is biased due to the inter-domain discrepancy, the manifold structure is difficult to be constructed. Therefore, we propose a new feature named Histogram Feature of Neighbors (H-FON) from the sample statistics perspective. Extensive experiments on various benchmark databases validate that our method outperforms many state-of-the-art image retrieval methods for domain adaptive image retrieval. The source code is available at https://github.com/fuxianghuang1/PWCF.

1. Introduction

The problem of domain adaptive image retrieval including single-domain retrieval and cross-domain retrieval is an important task for many practical applications. Single-
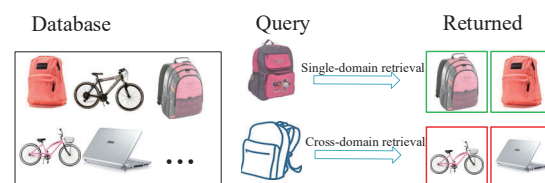

Figure 1. Illustration of our motivation. Many advanced methods have achieved excellent performance in solving the single-domain retrieval problem, but the performance drops significantly when they are used in cross-domain retrieval. In practice, queries and databases usually come from different domains, so it is necessary to solve the problem of cross-domain retrieval.

domain retrieval refers to a sort of image retrieval problem that the queries and databases are both from the same domain. On the contrary, cross-domain retrieval means the queries and databases permitting to come from different domains, which is more flexible and applicable in real-world applications. In practice, retrieval databases have often taken in ideal illumination/pose/background/camera conditions and queries usually obtained in uncontrolled conditions, which leads to the large discrepancy between the databases and queries. For example, mobile product image search [5] aims at identifying a product, or retrieving similar products from the online shopping domain based on a photo captured in unconstrained scenarios by a mobile phone camera.

However, as shown in Fig. 1, most of the existing methods only focus on single-domain retrieval and the performance deteriorates rapidly in handling cross-domain retrieval. Few people have proposed solutions to cross-domain retrieval problems. DARN [10] simultaneously integrates the attributes and visual similarity constraint into the retrieval feature learning to solve the cross-domain retrieval problem. However, attributes are usually insufficient and sorting high-dimensional features requires a lot of computation, resulting in slow retrieval.

Recently, due to the low storage and high computation efficiency of binary codes, the hashing algorithm has been
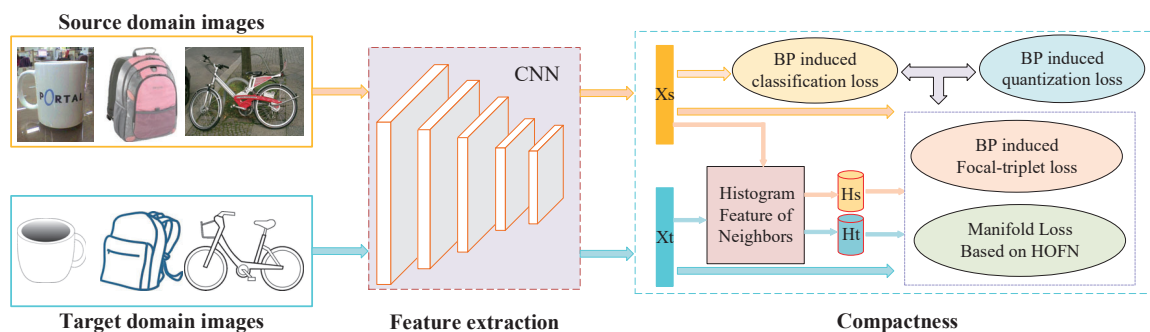
Figure 2. Diagram of PWCF, which includes four parts: 1) BP induced focal-triplet loss, 2) BP induced classification loss, 3) BP induced quantification loss and 4) manifold loss based on Histogram Feature of Neighbors.

widely used for many applications [6, 8, 13, 15, 24, 26, 30, 33, 36]. Hashing aims to map high-dimensional content features of samples into Hamming space (binary space) and generate a set of low-dimensional binary codes to represent samples. In consequence, the cost of data storage can be largely reduced, thus the retrieval speed can be improved with the Hamming distance using binary operation (XOR).

However, most of the existing hashing methods [14, 16, 18, 19, 21, 23, 29, 34] assume that the distributions of retrieval databases and queries are similar while ignoring the inter-domain discrepancy, which makes them difficult to accurately capture correlations between cross-domain samples. Consequently, although most of the existing hashing methods have achieved significant performance for single-domain retrieval, they perform poorly when queries and databases come from different domains.

To address the above problem, we propose an effective domain adaptive image retrieval method named Probability Weighted Compact Feature Learning (PWCT), which takes into account the similarity/dissimilarity relation between the samples from different domains to learn compact binary feature representations. Inspired by transfer learning (TL) [27], we transfer knowledge across different domains to leverage knowledge between different domains and explore cross-domain sample correlations. Our goal is to use the available labeled data as a source domain to help us learn the projection matrix and get more discriminant binary codes. Instead of simply adding source domain data to expand the training set for better retrieval in the target domain, different from existing transfer hashing methods [37, 20], we focus on exploring the correlation of samples and data distribution discrepancy between domains to achieve good performance on cross-domain retrieval. To improve the performance of cross-domain retrieval, we propose our loss functions from a Bayesian Perspective. Specifically, we derive our loss functions: BP induced focal-triplet loss, BP induced quantization loss and BP induced classification loss by seeking for the Maximum A Posteriori Estimation (MAP) solution to promote the correlation between sam-

ples from different domains in Hamming space, ensure the discrimination of binary codes, and reduce the information error causing by quantification.

Besides, considering that the underlying manifold structure across different domains is extremely helpful to capture meaningful nearest neighbors correlation of different domains, we propose a common manifold to capture the inherent neighborhood structure in the source domain and target domain to further ensure that the correlation of different domains is preserved in Hamming space. However, the similarity between the samples from different domains is difficult to measure in terms of original content features. The same class samples from different domains may not be close, caused by inter-domain discrepancy. To handle such a problem, we consider the distribution characteristics of k nearest neighbors for each sample in respective domains and propose a new statistical feature called Histogram Feature of Neighbors (HFON) from the perspective of sample statistics to reduce the influence of data distribution discrepancy between domains. The main contributions and novelties of this paper are summarized as follows:

- In this paper, we propose an effective domain adaptive image retrieval method named Probability Weighted Compact Feature Learning (PWCF) to achieve fast and accurate retrieval. Fig. 2 shows the framework of our PWCF. To the best of our knowledge, we are the first to propose a new and practical adaptive cross-domain retrieval problem.

- We propose loss functions named BP induced focal-triplet loss, BP induced quantization loss, and BP induced classification loss which seeks for the Maximum A Posteriori Estimation (MAP) solution to explore the similarity/dissimilarity between samples from different domains, ensure the discrimination, and reduce the information error.

- In our PWCF, we propose a Histogram Feature of Neighbors (HFON) from the perspective of statistics to reduce the influence of the domain disparity and a

common manifold structure based on HFON to further preserve the correlation between samples from different domains.

- Extensive experiments on various benchmark databases have been conducted. The experimental results verify that our method outperforms many state-of-the-art image retrieval methods for both cross-domain retrieval and single domain retrieval.

2. Probability Weighted Compact Feature Learning

2.1. Notations and Definitions

Suppose that we have n_t target samples unlabeled $\mathbf{X}_t = \{\mathbf{x}_{t_i}\}_{i=1}^{n_t} \in \mathbb{R}^{d \times n_t}$ and n_s source samples labeled $\mathbf{X}_s = \{\mathbf{x}_{s_i}\}_{i=1}^{n_s} \in \mathbb{R}^{d \times n_s}$. $\mathbf{Y}_s = \{\mathbf{y}_{s_i}\}_{i=1}^{n_s} \in \mathbb{R}^{c \times n_s}$, where $\mathbf{y}_{s_i} \in \mathbb{R}^{c \times 1}$ is the label vector, of which the maximum item indicates the assigned class of $\mathbf{x}_{s_i}$. We denote $\mathbf{X} = [\mathbf{X}_t, \mathbf{X}_s]$ and $n = n_t + n_s$. We aim to learn a set of compact binary codes $\mathbf{B}_t = \{\mathbf{b}_{t_i}\}_{i=1}^{n_t} \in \{-1, 1\}^{r \times n_t}$ and $\mathbf{B}_s = \{\mathbf{b}_{s_i}\}_{i=1}^{n_s} \in \{-1, 1\}^{r \times n_s}$ to represent the samples, where $\mathbf{b}_{t_i}$ is the corresponding binary codes of $\mathbf{x}_{t_i}$ and $\mathbf{b}_{s_i}$ is the corresponding binary codes of $\mathbf{x}_{s_i}$. d and r represent the original content feature dimension of each sample and the length of binary codes, respectively. In PWCF, both the data of the source domain and the target domain are used to learn a projection $\mathbf{W} \in \mathbb{R}^{d \times r}$. Then, the r-dimensional feature (i.e., the continuous real values of binary codes) is denoted as $f_i = \mathbf{W}^\top \mathbf{x}_i$. The binary codes are quantified as $\mathbf{b}_i = sgn(f_i) \in \{-1, 1\}^{r \times 1}$. Here sgn($v$) is the sign function, which returns 1 if $v \geq 0$ and -1 otherwise. In this paper, $\| \cdot \|$ is the ℓ_2 norm for vectors and Frobenius norm for matrices.

2.2. Compactness: A Bayesian Perspective

In order to achieve higher accuracy, we hope to explore the correlation of different samples. Given a triplet $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) \in \mathbf{X}$, let s_{ij} represent the pair-wise similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. $s_{ij} = 1$ means they have the same label. Instead, $s_{ij} = 0$ means they have different labels.

Without loss of generality, let $p(f_i, f_j, f_k | s_{ij}, s_{ik})$ be the posterior probability of feature representation f_i, f_j, f_k for triplet sample set $\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k$. Here we suppose that f_i, f_j, f_k are the r-dimensional features of samples $\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k$, respectively. With the assumption of conditional independence of each pair and Bayesian formulation, the joint posterior probability density function of the triplet training set can be generally represented as:

$$\prod_{i,j,k \in \mathbf{X}} p(f_i, f_j, f_k | s_{ij}, s_{ik}) \Leftrightarrow$$
$$\prod_{i,j,k \in \mathbf{X}} p(s_{ij}, s_{ik} | f_i, f_j, f_k) \, p(f_i) \, p(f_j) \, p(f_k) \quad (1)$$

where $p(s_{ij}, s_{ik} | f_i, f_j, f_k)$ is the likelihood probability and $p(f_i)$, $p(f_j)$ and $p(f_k)$ are the prior probability of r-dimensional feature. We suppose that the likelihood probability density function to be exponential distribution, considering that the exponential distribution has shown fast convergence to a stable state. Let $d_{ij} = \|f_i - f_j\|^2$ and $d_{ik} = \|f_i - f_k\|^2$. Considering the sample pair similarity, the likelihood probability density function is expressed as:

$$p(s_{ij}, s_{ik} | f_i, f_j, f_k) =$$
$$\begin{cases} e^{-|d_{ij} - d_{ik} + m|}, & if \ s_{ij} = 1, s_{ik} = 0 \\ e^{-|-d_{ij} + d_{ik} + m|}, & if \ s_{ij} = 0, s_{ik} = 1 \\ e^{-|d_{ij} + d_{ik}|}, & if \ s_{ij} = 1, s_{ik} = 1 \\ e^{-|-d_{ij} - d_{ik}|}, & if \ s_{ij} = 0, s_{ik} = 0 \end{cases} \quad (2)$$
$$= e^{-|(-1)^{\bar{s}_{ij}} d_{ij} + (-1)^{\bar{s}_{ik}} d_{ik} + \alpha \cdot m|}$$

where $\bar{s}_{ij} = -s_{ij}$, $\alpha = s_{ij} \oplus s_{ik}$ and $\oplus$ is the XOR operation. m is the margin. The purpose of this setting is to make the samples of the same class closer and the samples of different classes farther.

We aim to seek the solution of Maximum A Posteriori Estimation (MAP) of Eq. (1) from the Bayesian perspective. To mitigate the influence of the likelihood probability of hard pairs on the posterior probability maximization, we add a modulating factor ω_{ijk} to the likelihood probability where $\omega_{ijk} = (1 - p(s_{ij}, s_{ik} | f_i, f_j, f_k))^\gamma$ and $\gamma \geq 0$. In other words, the modulating factor reduces the contribution of easy pairs and penalizes more on those hard pairs. For convenience, the same (different) labeled samples which have a large (small) distance are named as hard pairs, and the same (different) labeled samples that have a small (large) distance are named as easy pairs.

Besides, considering the quantization loss and discrimination of binary code, the prior probability is written as $p(f_i) = e^{-\theta d(\mathbf{b}_i, f_i)} \cdot e^{-\lambda_1 d(\mathbf{y}_i, \mathbf{C}^\top \mathbf{b}_i)} \cdot e^{-\lambda_2 \|\mathbf{C}\|^2}$, where θ, λ_1 and λ_2 are hyper-parameters. $\mathbf{C}$ is a classifier and we will discuss the detail later in classification loss. By taking the natural logarithm, our objective function is written as:

$$\max \sum_{i,j,k \in \mathbf{X}} \omega_{ijk} \log p(s_{ij}, s_{ik} | f_i, f_j, f_k) +$$
$$\sum_{i \in \mathbf{X}} \log p(f_i) + \sum_{j \in \mathbf{X}} \log p(f_j) + \sum_{k \in \mathbf{X}} \log p(f_k) \quad (3)$$

In optimization, we consider the case that positive pairs and negative pairs both exist. In order to construct triplets, we can set $\mathbf{x}_i$ as the anchor, $\mathbf{x}_j$ is similar to the anchor, and $\mathbf{x}_k$ is dissimilar to the anchor. Then Eq. (3) is

$$\min \sum_{i,j,k \in \mathbf{X}} (1 - e^{[d_{ij} - d_{ik} + m]_+})[d_{ij} - d_{ik} + m]_+$$
$$+ \theta \sum_{i \in \mathbf{X}} d(\mathbf{b}_i, f_i) + \lambda_1 \sum_{i \in \mathbf{X}} d(y_i, \mathbf{C}^\top \mathbf{b}_i) + \lambda_2 \|\mathbf{C}\|^2 \quad (4)$$

where $[x]_+$ denotes the operator of $\max(x,0)$, which makes sure $p(s_{ij}, s_{ik}|f_i, f_j, f_k) \in (0,1]$ and improves the convergence. Clearly, if without $\max(x,0)$, the probability of the exponential probability in Eq.(2) may be larger than 1. So, in Eq.(4), the $\max(x,0)$ is naturally resulted with clear probabilistic interpretation.

BP induced Focal-triplet loss. The first term in Eq. (3) is a variant of standard triplet loss named BP induced focal-triplet loss. If we enumerate all the sample pairs, it will take a lot of time for training. So we will just pick some cross-domain triplets, which are more effective at promoting inter-domain correlations. In other words, we construct cross-domain triplets before training. Specifically, for each sample, if it comes from the source domain, we select a positive sample and a negative sample from the target domain. Otherwise, if it is from the target domain, we select a positive sample and a negative sample from the source domain. Since there is no label in the target domain, we first use the source domain data to predict the pseudo-label of the target domain by the KNN algorithm. We can get n cross-domain triplets. For ease of understanding, let's $(\mathbf{x}_i^g, \mathbf{x}_{i,p}^{\overline{g}}, \mathbf{x}_{i,n}^{\overline{g}}), i \in [1, n]$ represent all selected triplets, where g and $\overline{g}$ come from different domain. If $g \in \mathbf{X}_s$, then $\overline{g} \in \mathbf{X}_t$. Otherwise, if $g \in \mathbf{X}_t$, then $\overline{g} \in \mathbf{X}_s$. Then, the BP induced focal-triplet loss can be written as:

$$\mathcal{T}ri = \sum_{i,j,k \in \mathbf{X}} (1 - e^{[d_{ij} - d_{ik} + m]_+})[d_{ij} - d_{ik} + m]_+ =$$
$$\sum_{i=1}^{N} \omega_i \left[\left\| \mathbf{W}^\top \mathbf{x}_i^g - \mathbf{W}^\top \mathbf{x}_{i,p}^{\overline{g}} \right\|^2 - \left\| \mathbf{W}^\top \mathbf{x}_i^g - \mathbf{W}^\top \mathbf{x}_{i,n}^{\overline{g}} \right\|^2 + m \right]_+ \tag{5}$$

where ω_i is the weight of ith group selected triplet and $\omega_i = \left(1 - e^{-\left(\left\| \mathbf{W}^\top \mathbf{x}_i^g - \mathbf{W}^\top \mathbf{x}_{i,p}^{\overline{g}} \right\|^2 - \left\| \mathbf{W}^\top \mathbf{x}_i^g - \mathbf{W}^\top \mathbf{x}_{i,n}^{\overline{g}} \right\|^2 + m\right)}\right)^\gamma$.
As shown in Fig. 3, the focal-triplet loss, which is a variant of standard triplet loss, imposes different importance for different triplets by down-weighting easy pairs and up-weighting hard pairs. In the training phase, we choose the hard triplets that satisfy the maximization of intra-class distance and the minimization of inter-class distance to improve the training speed. Considering the data distribution discrepancy in different domains, the Euclidean distance of the original content feature extracted from different domains may not measure the similarity of samples. So we use the Histogram Feature of Neighbors rather than the original content feature to calculate the distance of different samples across domains. The Histogram Feature of Neighbors will be explained in detail in the next section.

BP induced quantization loss. The second term in Eq. (4) is named BP induced quantization loss, which aims to reduce quantization error between binary codes and low-dimensional feature representation obtained by map-

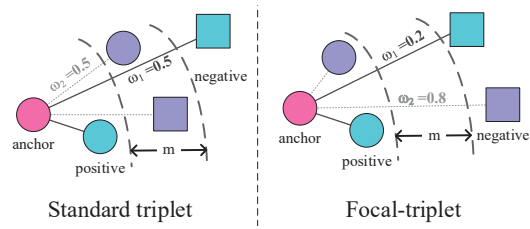Standard triplet | Focal-triplet

Figure 3. Illustration of the proposed BP induced focal-triplet loss: Standard triplet makes the positive sample closer to the anchor and the negative sample further away from the anchor by the same force (weight). However, positive samples and negative samples may not be separated for hard pairs in this case, which results in training instability. To address it, our BP induced focal-triplet loss can down-weight easy pairs and up-weight hard pairs so as to the distance to the cross-domain positives can be minimized and the distance to the cross-domain negatives can be maximized.

ping (i.e., the continuous real values of binary codes). The BP induced quantization loss can be formulated as:

$$\mathcal{Q} = \sum_{i \in \mathbf{X}} d(\mathbf{b}_i, f_i) = \sum_{i=1}^{n} \left\| \mathbf{b}_i - \mathbf{W}^\top \mathbf{x}_i \right\|^2$$
$$= \left\| \mathbf{B}_t - \mathbf{W}^\top \mathbf{X}_t \right\|^2 + \left\| \mathbf{B}_s - \mathbf{W}^\top \mathbf{X}_s \right\|^2 \tag{6}$$

BP induced classification loss. The third term in Eq. (4) is named BP induced classification loss. Inspired by SDH [28], we consider that good binary codes should be with good discrimination. We take advantage of the label information to train a classifier $\mathbf{C}$ and $\mathbf{C}^\top \mathbf{b}_i$ represent the predicted label of the ith sample. We want to use binary codes to predict labels that are as authentic as possible. In this paper, to avoid the negative impact of pseudo labels, we only use the source domain sample when we calculate the classification loss. The BP induced classification loss can be formulated as:

$$\mathcal{C} = \sum_{i \in \mathbf{X}} d(y_i, \mathbf{C}^\top \mathbf{b}_i) = \sum_{i=1}^{n} \left\| \mathbf{y}_i - \mathbf{C}^\top \mathbf{b}_i \right\|^2$$
$$\approx \left\| \mathbf{Y}_s - \mathbf{C}^\top \mathbf{B}_s \right\|^2 \tag{7}$$

The regularization $\|\mathbf{C}\|^2$, i.e., the last term in Eq. (3), is used to avoid trivial solution and overfitting.

2.3. Manifold Loss Based on Histogram Feature of Neighbors

We argue that the nearest-neighbor relationship of samples in a single domain is regular. In other words, if two samples from different domains are similar, the classes of their neighbors in their respective domains should be similar. Based on this assumption, we propose a statistical feature named Histogram Feature of Neighbors (HFON) to reduce the domain disparity. Specifically, we use $\mathbf{h}_i \in \mathbb{R}^{c \times 1}$
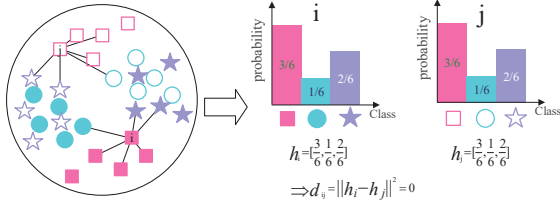
Figure 4. Illustration of the proposed Histogram Feature of Neighbors (HFON): We use different shapes to represent samples of different classes, The samples from the source domain and target domain are represented by solid and hollow shapes, respectively. i and j belong to the same class, but the distance is far. To measure the similarity of cross-domain samples more accurately, we proposed a Histogram Feature of Neighbors according to the neighbor relationship in a domain. For example, we find 6 nearest neighbors of i and j in their respective domains and calculate the probability of each class of these nearest-neighbor samples. H-FON vector is made up of these probabilities. Then the HFON distances of similar samples in different domains are close.

to represent the HFON vector of $\mathbf{x}_i$ and c is the number of classes. We find k nearest neighbors of each sample in their respective domains and calculate the probability of each class of these nearest-neighbor samples. The ath element of the HFON can be written as $\mathbf{h}_{i_a} = \frac{n_{i_a}}{k}$ where $a \in [1, c]$, n_{i_a} represents the total number of samples belonging to class c in the k nearest neighbors of the ith sample. Fig. 4 shows the details of the Histogram Feature of Neighbors.

The underlying manifold structure across different domains, which is extremely helpful to capture meaningful nearest neighbors of different domains. Therefore, we want to keep the common manifold structure by taking advantage of local similarities. To minimize the representation error of the low-dimensional features between different neighbor samples, the manifold loss can be written as:

$$\mathcal{M} = \sum_{i,j \in \mathbf{X}} \|f_i - f_j\|^2 Z_{ij}$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{n} \left\| \mathbf{W}^\top \mathbf{x}_i - \mathbf{W}^\top \mathbf{x}_j \right\|^2 Z_{ij} \quad (8)$$

$$= \mathbf{W}^\top \mathbf{X} \mathbf{L} \mathbf{X}^\top \mathbf{W}$$

Similar to LPP [9], $\mathbf{L} = \mathbf{D} - \mathbf{Z}$ is the Laplacian matrix and $D_{ii} = \sum_j Z_{ij} = \sum_i Z_{ij}$. Here $\mathbf{Z}$ is a sparse symmetric $n \times n$ matrix with Z_{ij} having the weight of the edge connecting $\mathbf{x}_i$ and $\mathbf{x}_j$, and 0 if there is no such connection. To reduce the domain disparity, let $Z_{ij} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}}$ when $\mathbf{x}_i$ and $\mathbf{x}_j$ come from the same domain. Otherwise, $Z_{ij} = e^{-\frac{\|\mathbf{h}_i - \mathbf{h}_j\|^2}{\sigma^2}}$ when $\mathbf{x}_i$ and $\mathbf{x}_j$ come from different domains where $\mathbf{h}_i$ and $\mathbf{h}_j$ denote the HFON with respect to $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively.

Overall objective function. Finally, the overall objec-

tive function is rewritten as:

$$\min_{\mathbf{W}, \mathbf{C}, \mathbf{B}_t, \mathbf{B}_s} \mathcal{T}ri + \theta \mathcal{Q} + \lambda_1 \mathcal{C} + \lambda_2 \|\mathbf{C}\|^2 + \lambda_3 \mathcal{M}$$

$$s.t. \mathbf{W}^\top \mathbf{W} = \mathbf{I}, \mathbf{B}_t = sgn(\mathbf{W}^\top \mathbf{X}_t), \mathbf{B}_s = sgn(\mathbf{W}^\top \mathbf{X}_s)$$
$$(9)$$

where the constraint, $\mathbf{W}^\top \mathbf{W} = \mathbf{I}$, is used to make $\mathbf{W}$ be orthogonal projections in order to guarantee the discrimination of binary codes.

2.4. Optimization

In this paper, we adopt an alternating optimization procedure to iteratively optimize $\mathbf{W}$, $\mathbf{C}$, $\mathbf{B}_t$ and $\mathbf{B}_s$. As the non-convex sgn(·) function makes Eq. (8) a NP-hard problem, we relax the sgn(x) function as its signed magnitude x [20].

W-Step. Given $\mathbf{C}$, $\mathbf{B}_t$ and $\mathbf{B}_s$, updating $\mathbf{W}$ is a typical optimization problem with orthogonality constraints. Let $\mathbf{G}$ be the partial derivative of the objective function Eq. (9) with respect to $\mathbf{W}$ and $\mathbf{G}$ is represented as:

$$\mathbf{G} = 2 \sum_{i \in \mathcal{T}^+} \omega_i ((\mathbf{x}_i^g - \mathbf{x}_{i,p}^{\overline{g}})(\mathbf{x}_i^g - \mathbf{x}_{i,p}^{\overline{g}})^\top - (\mathbf{x}_i^g - \mathbf{x}_{i,n}^{\overline{g}})(\mathbf{x}_i^g - \mathbf{x}_{i,n}^{\overline{g}})^\top) \mathbf{W}$$

$$+ 2\theta (\mathbf{X}_t \mathbf{X}_t^\top \mathbf{W} - \mathbf{X}_t \mathbf{B}_t^\top + \mathbf{X}_s \mathbf{X}_s^\top \mathbf{W} - \mathbf{X}_s \mathbf{B}_s^\top)$$

$$+ 2\lambda_3 \mathbf{X} \mathbf{L} \mathbf{X}^\top \mathbf{W}$$
$$(10)$$

where $\mathcal{T}^+$ contains all selected cross-domain triplets and $\left\| \mathbf{W}^\top \mathbf{x}_i^g - \mathbf{W}^\top \mathbf{x}_{i,p}^{\overline{g}} \right\|^2 - \left\| \mathbf{W}^\top \mathbf{x}_i^g - \mathbf{W}^\top \mathbf{x}_{i,n}^{\overline{g}} \right\|^2 + m \geq 0$. Based on the orthogonal constraint optimization procedure in [35], we can define a skew-symmetric matrix [1] as $\mathbf{A} = \mathbf{G} \mathbf{W}^\top - \mathbf{W} \mathbf{G}^\top$. Then, we adopt Crank Nicolson like scheme to update the orthogonal matrix $\mathbf{W}$

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \frac{\tau}{2} \mathbf{A} \left(\mathbf{W}^{(t)} + \mathbf{W}^{(t+1)} \right) \quad (11)$$

where τ denotes the step size. We empirically set $\tau = 0.1$. By solving Eq. (11), we can get

$$\mathbf{W}^{(t+1)} = \mathbf{Q} \mathbf{W}^{(t)} \quad (12)$$

and $\mathbf{Q} = (\mathbf{I} + \frac{\tau}{2} \mathbf{A})^{-1} (\mathbf{I} - \frac{\tau}{2} \mathbf{A})$. We iteratively update $\mathbf{W}$ several times based on Eq. (12) with the Barzilai-Borwein (BB) method [35]. In addition, please note that when iteratively optimizing $\mathbf{W}$, the initial $\mathbf{W}$ is set to be the updated one in the last round. For the first round, $\mathbf{W}$ is initialized by PCA.

C-Step. Given $\mathbf{W}$, $\mathbf{B}_t$ and $\mathbf{B}_s$, taking the partial deviation of the objective function with respect to $\mathbf{C}$ to be zero, we derive

$$\mathbf{C} = \left(\lambda_1 \mathbf{B}_s \mathbf{B}_s^\top + \lambda_2 \mathbf{I} \right)^{-1} \lambda_1 \mathbf{B}_s \mathbf{Y}_s^\top \quad (13)$$

$\mathbf{B}_t$-Step. Given $\mathbf{W}$, $\mathbf{C}$ and $\mathbf{B}_s$, by relaxing the sign function $sgn(.)$, the solution can be obtained

$$\mathbf{B}_t = sgn(\mathbf{W}^\top \mathbf{X}_t) \quad (14)$$

B$_s$-Step. Given $\mathbf{W}$, $\mathbf{C}$ and $\mathbf{B}_t$, we obtain the approximate solution for hash codes by relaxing the sign function.

$$\mathbf{B}_s = sgn\left((\theta\mathbf{I} + \lambda_1\mathbf{C}\mathbf{C}^\top)^{-1}(\theta\mathbf{W}^\top\mathbf{X}_s + \lambda_1\mathbf{C}\mathbf{Y}_s)\right) \tag{15}$$

The details of the proposed algorithm are described in Algorithm 1.

Algorithm 1 PWCF learning

Input: Training samples $\mathbf{X} = [\mathbf{X}_t, \mathbf{X}_s]$ and source labels $\mathbf{Y}_s$; code length r; maximum iteration times T; number of neighbors k; parameters θ, λ_1, λ_2 and λ_3

Output: $\mathbf{W}$, $\mathbf{C}$, $\mathbf{B}_t$ and $\mathbf{B}_s$

1: Obtain the pseudo-labels of target domain by KNN trained with $(\mathbf{X}_s,\mathbf{Y}_s)$.
2: Calculate the HFON of target domain and source domain, respectively.
3: Construct cross-domain triples: $(\mathbf{x}_i^g, \mathbf{x}_{i,p}^{\overline{g}}, \mathbf{x}_{i,n}^{\overline{g}}), i \in [1, N]$.
4: Initialize $\mathbf{W}$ as the top r eigenvectors of $\mathbf{X}\mathbf{X}^\top$ by PCA. $\mathbf{B}_t$ and $\mathbf{B}_s$ are random binary matrices, respectively.
5: Loop until converge or reach maximum iterations:
 W-Step. update $\mathbf{W}$ by solving Eq. (12);
 C-Step. update $\mathbf{C}$ by solving Eq. (13);
 B$_t$-Step. update $\mathbf{B}_t$ by solving Eq. (14);
 B$_s$-Step. update $\mathbf{B}_s$ by solving Eq. (15).

Computation Complexity: Since the hard triplets and the Laplacian matrix can be pre-computed, the total computation cost of our PWCF in Algorithm 1 is $\mathcal{O}(T(n(d^2r + dr) + 3d^2r + n_sr^2c + n_tdr + n_s(r^2c^2 + r^2cd)))$ and linear to the number of samples, where $n = n_s + n_t$. In practice, T, d, r and c will be much less than n. Hence, the binary codes learning is efficient.

3. Experiment

3.1. Experimental Settings

Datasets: We perform the experiments on four groups benchmark datasets:

- The **MNIST** [17] and USPS [11] are two famous digital datasets sharing ten handwritten digits from 0 to 9. According to [25], each image is resized to 16×16. We use the MNIST as the source domain and USPS is used as the target domain in the *MNIST&USPS* dataset.

- The **VLCS** [31] dataset aggregates photos from Caltech101, LabelMe, Pascal VOC2007 and SUN09, which provides a 5-way multi-class benchmark on five common classes: bird, car, chair, dog and person. In our experiments, every image is represented by a 4096-d CNN feature vector [4]. We use VOC2007 dataset

including 3376 images as the source domain and Caltech101 dataset containing 1415 images is used as the target domain in the *VOC2007&Caltech101* dataset.

- The **Cross-dataset Testbed** [3] is a Decaf7 based cross-dataset image classification dataset, which contains 40 categories of images from 3 domains: 3,847 images in Caltech256, 4,000 images in ImageNet, and 2,626 images for SUN. In our experiments, each image is represented by a 4096-d CNN feature vector [4]. Caltech256 is used as the source domain and ImageNet is used as the target domain in the *Caltech256&ImageNet* dataset.

- The **Office-Home** dataset [32] consists of images from 4 different domains: Artistic images (i.e., paintings, sketches and/or artistic depictions), Clip Art images, Product images without background and Real-World images (i.e., regular images captured with a camera). For each domain, the dataset contains images of 65 object categories found typically in Office and Home settings. In our experiments, each image is represented as a 4096-d feature by VGG-16. Each domain is used as the source domain and the target domain, respectively.

Implementation details: We choose eleven state-of-the-art hashing methods, including SH [34], ITQ [7], DSH [14], LSH [2], SGH [12], OCH [19], GTH [20], ITQ+ [37], LapITQ+ [37], KSH [22] and SDH [28] as baselines. We use the public codes and suggested parameters of these methods from the corresponding authors. For our P-WCF, we empirically set θ to 1e2, λ_1 to 1, λ_2 to 1e3 and λ_3 to 1e4. For unsupervised methods, we use all the training samples including source domain and target domain in the training phase. For a fair comparison, we introduce a NoTL method that only uses the target domain to train the ITQ model. For supervised methods, we use the training samples and labels of source domain to train. All of the methods use identical training sets and testing sets. Specifically, for each dataset, we randomly select 500 images of the target domain images as a testing set (queries) and rest images as a training set. In the testing phase, class labels are used to determine whether a sample returned for a given query is considered a true positive. Moreover, the widely used criterion, i.e., mean average precision (MAP), is used as the performance metric. To remove the randomness for sampling, we repeat each algorithm 10 times and report their mean of MAP. We also show the precision and recall curves.

To verify the performance of our method in the scenarios of domain adaptive retrieval, we report the retrieval performance including cross-domain retrieval and single-domain retrieval on *MNIST&USPS*, *VOC2007&Caltech101*, and *Caltech256&ImageNet* databases when the code length is set as 16, 32, 48, 64, 96 and 128, respectively. For cross-domain retrieval, the training samples from the source do-

Table 1. The MAP scores (%) on MNIST&USPS, VOC2007&Caltech101, and Caltech256&ImageNet databases with varying code length from 16 to 128 for cross-domain retrieval.

Bit	MNIST&USPS						VOC2007&Caltech101						Caltech256&ImageNet					
	16	32	48	64	96	128	16	32	48	64	96	128	16	32	48	64	96	128
NoTL	28.13	30.05	28.24	30.34	31.76	31.72	35.95	37.86	38.28	38.49	38.67	38.97	15.10	19.77	22.80	24.39	26.07	27.28
SH	15.71	13.85	12.05	11.78	11.38	11.78	29.94	30.26	32.51	33.76	32.59	33.03	10.37	11.67	12.17	11.88	12.67	12.89
ITQ	27.38	30.92	31.44	32.25	33.12	33.44	40.13	39.63	39.45	39.98	39.27	39.89	16.94	22.00	24.44	26.21	27.96	28.89
DSH	21.15	27.53	29.71	26.13	26.60	28.94	40.97	42.03	43.06	45.81	43.78	42.86	8.27	9.60	11.55	12.34	13.56	15.64
LSH	16.25	16.99	23.23	20.38	19.70	26.98	33.40	33.99	34.03	32.89	34.12	34.50	5.36	6.72	10.39	12.71	15.60	17.08
SGH	24.83	24.78	25.85	27.78	28.26	29.35	35.77	34.06	33.60	33.11	32.75	32.41	12.49	17.23	20.34	21.75	24.46	25.42
OCH	18.94	25.73	26.73	26.34	27.88	29.22	71.50	72.27	72.65	72.71	69.17	68.91	11.56	15.36	17.49	20.18	22.00	22.90
ITQ+	20.27	20.53	16.77	15.87	17.79	14.90	35.35	34.48	34.33	34.42	34.05	34.74	–	–	–	–	–	–
LapITQ+	26.38	26.31	24.91	24.61	22.04	21.33	38.95	38.43	39.64	39.35	39.33	38.76	–	–	–	–	–	–
GTH	19.10	24.17	24.27	24.38	23.64	29.36	36.70	38.95	37.23	37.87	37.70	38.36	11.56	14.79	16.97	19.53	20.88	22.38
KSH	43.75	46.91	50.02	47.43	45.25	46.81	74.74	76.05	76.71	76.70	76.22	73.14	20.34	12.07	26.77	32.83	35.28	34.49
SDH	29.98	43.02	42.57	46.56	42.40	48.12	67.60	65.75	68.58	65.06	65.66	67.03	18.05	25.71	26.23	26.38	26.77	26.29
PWCF	47.47	51.99	51.44	51.75	50.89	53.95	79.38	80.42	79.24	79.31	78.15	78.87	22.46	30.58	35.29	35.24	38.92	40.32

Table 2. The MAP scores (%) on MNIST&USPS, VOC2007&Caltech101, and Caltech256&ImageNet databases with varying code length from 16 to 128 for single-domain retrieval.

Bit	MNIST&USPS						VOC2007&Caltech101						Caltech256&ImageNet					
	16	32	48	64	96	128	16	32	48	64	96	128	16	32	48	64	96	128
NoTL	67.22	69.31	70.52	70.78	71.64	71.88	98.12	98.13	98.36	98.36	98.58	98.82	15.12	20.50	22.66	24.22	25.954	27.43
SH	47.07	49.19	49.24	49.64	49.69	49.03	66.56	64.44	66.39	68.39	67.29	66.12	10.44	11.33	12.24	12.24	12.85	13.51
ITQ	63.37	69.96	69.53	70.19	71.22	71.59	99.03	99.15	99.09	99.10	99.14	99.19	15.96	20.57	23.07	24.16	26.39	27.63
DSH	45.76	54.23	58.03	59.92	61.80	63.50	94.58	93.88	95.93	97.07	97.93	97.86	8.09	10.21	11.83	12.89	15.43	15.15
LSH	47.21	55.63	59.81	60.54	60.81	62.90	61.17	65.69	79.25	84.76	88.81	88.19	5.15	6.87	10.22	12.63	15.37	17.38
SGH	58.41	63.61	64.69	65.79	66.55	66.81	86.06	86.70	88.49	88.68	91.18	91.71	12.37	16.75	19.54	20.90	23.37	24.93
OCH	53.56	58.49	60.48	63.96	66.29	65.39	89.36	97.47	98.30	98.45	98.71	99.21	10.11	15.05	17.47	19.58	20.70	22.31
ITQ+	41.87	37.94	37.00	37.23	35.19	34.88	64.15	59.00	56.94	56.12	54.74	52.61	–	–	–	–	–	–
LapITQ+	54.21	55.64	53.66	52.58	51.56	49.80	70.53	69.08	67.95	66.32	69.49	67.59	–	–	–	–	–	–
GTH	53.20	58.78	62.17	63.23	62.94	60.59	90.25	82.72	92.93	93.73	94.33	87.44	11.73	15.02	17.67	19.21	20.87	21.90
KSH	26.06	37.11	42.57	41.89	40.89	38.07	96.98	91.90	93.03	92.58	96.27	95.86	16.20	11.23	19.88	25.64	28.07	29.58
SDH	50.32	54.20	57.29	57.48	60.64	60.76	88.49	86.54	88.27	87.28	89.11	89.24	13.92	18.72	21.20	21.64	22.74	23.94
PWCF	69.37	70.70	70.94	71.64	73.51	73.89	99.67	99.61	99.77	99.66	99.33	99.58	21.96	26.56	28.75	29.91	32.92	34.96

main are used as retrieval database. For the single-domain retrieval, the training samples from the target domain are used as retrieval database.

To further prove our versatility and cross-domain retrieval performance, a large number of experiments were carried out on *Office-Home*. For the sake of simplicity, Artistic images, Clip Art images, Product images, and Real-World images are replaced as A, C, P and R, respectively. A→C implies Artistic is the source domain and Clip Art is the target domain.

3.2. Experimental Results

In Table 1, we report the MAP scores (%) of all the compared methods and our method PWCF on *MNIST&USPS*, *VOC2007&Caltech101*, and *Caltech256&ImageNet* for cross-domain retrieval. Obviously, our PWCF outperforms compared methods on all databases in most cases. To further prove the effectiveness of our method, we conduct an experimental evaluation of single-domain retrieval. The results are shown in Table 2. We can see that our method is superior to the compared methods in both cross-domain retrieval and single-domain retrieval.

In Table 3, we report the MAP scores (%) of all the compared methods and our PWCF with 64 bits on *Office-Home* for cross-domain retrieval. We can see however the source domain and target domains are set, our methods perform

Table 3. The MAP scores (%) on Office-home databases with 64 bits for cross-domain retrieval. P→R implies Product is the source domain and Real-World is the target domain.

|	P→R	R→P	C→R	R→C	A→R	R→A	average
NoTL	25.02	29.09	16.98	14.73	27.38	21.05	22.38
SH	15.01	16.82	10.38	8.32	14.52	12.41	12.91
ITQ	26.32	29.13	17.60	15.88	26.86	21.99	22.96
SDH	8.86	7.72	6.44	6.15	10.07	9.11	8.06
LSH	10.74	14.78	9.85	8.37	12.24	10.02	11.00
SGH	24.12	26.47	16.10	14.14	22.82	20.41	20.68
OCH	20.06	20.23	10.97	10.61	19.35	15.32	16.09
ITQ+	17.94	–	10.61	–	–	15.16	14.57
LapITQ+	15.94	–	11.72	–	–	13.52	13.83
GTH	19.40	22.80	12.27	12.03	20.98	16.47	17.33
KSH	32.02	34.42	21.56	18.51	25.87	20.04	25.40
SDH	25.75	27.90	15.97	16.72	32.06	22.79	23.53
PWCF	34.03	34.44	24.22	18.42	34.57	28.95	29.11
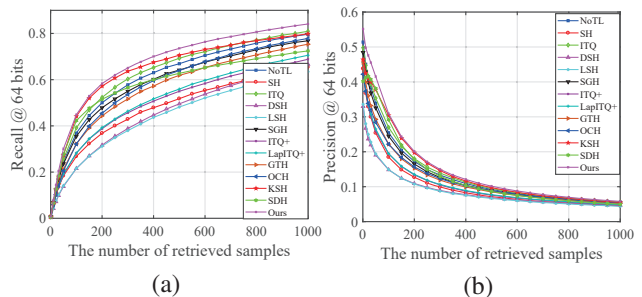
Figure 5. The influence of the number of retrieved samples for cross-domain retrieval on Product→Real with 64 bits.

Table 4. The MAP scores (%) on MNIST&USPS databases with varying code length from 16 to 128 for cross-domain retrieval or single-domain retrieval.

Bit	cross-domain						single-domain					
	16	32	48	64	96	128	16	32	48	64	96	128
PWCF	**47.47**	**51.99**	**51.44**	**51.75**	**50.89**	**53.95**	**69.37**	**70.70**	**70.94**	**71.64**	**73.51**	**73.89**
PWCF-T	45.13	48.76	50.02	50.57	50.66	51.43	51.45	62.18	68.25	69.91	71.62	72.81
PWCF-F	44.07	45.40	47.58	50.01	50.12	53.18	52.55	62.70	66.37	69.07	71.27	72.03
PWCF-M	29.13	30.76	33.44	34.56	35.68	36.22	62.16	67.84	69.04	70.25	70.68	71.32
PWCF-C	40.80	47.02	50.04	50.36	48.88	47.81	50.85	61.11	64.94	66.40	69.24	70.30
PWCF-H	29.09	32.14	34.89	36.36	36.84	36.88	49.12	64.13	67.82	69.15	70.64	71.09
PWCF-Q	10.89	10.65	10.64	10.60	10.63	10.52	15.22	15.43	16.24	16.83	19.05	18.49

better than others. The results certify that our PWCF has universality in practical application. We also show the influence of the number of retrieved samples for cross-domain retrieval task. Fig. 5 (a) shows the precision when the number of retrieved samples vary from 0 to 1000 and Fig. 5 (b) shows the recall when the number of retrieved samples vary from 0 to 1000. From the figures, we can see that our PWCF always presents competitive retrieval performance compared to baselines, which demonstrates the efficacy of our PWCF.

3.3. Convergence Analysis

The proposed PWCF is solved with a variable alternating strategy, and the convergence can be guaranteed. We present the convergence curves of the objective function in Fig. 6, from which we see that PWCF can quickly converge to an optimal solution within several iterations.
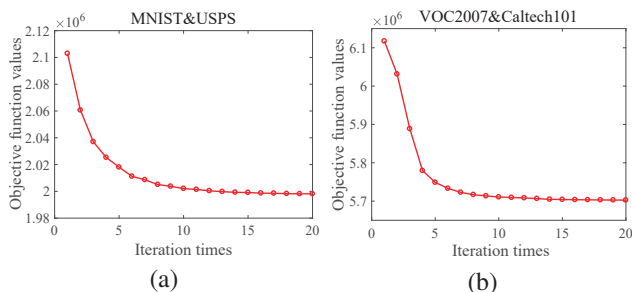

Figure 6. Convergence of Algorithm 1 on (a) MNIST&USPS and (b)VOC2007&Caltech101 with 64 bits.

3.4. Ablation Study

We investigate six variants of PWCF in Table 4: (1) PWCF-T is the PWCF variant without BP induced focal-triplet loss $\mathcal{T}ri$. (2) PWCF-F is the STH variant, which replaces the BP induced focal-triplet loss as the standard triplet loss. (3) PWCF-M is the PWCF variant without manifold loss $\mathcal{M}$. (4) PWCF-C is the PWCF variant without BP induced classifier loss $\mathcal{C}$. (5) PWCF-H is the PWCF variant without the Histogram Feature of Neighbors, which calculates the weight matrix $\mathbf{Z}$ by using original content features. Also, the hard triplets are constructed by original features

without the Histogram Feature of Neighbors. (6) PWCF-Q is the PWCF variant without BP induced quantization loss $\mathcal{Q}$. We report the results to different code lengths on the *MNIST&USPS* dataset for single-domain and cross-domain retrieval.

We can see that the four parts of our model have different effects on retrieval performance. Comparing PWCF-T, PWCF-F, PWCF, triplet loss is good for training PWCF and our proposed focal-triplet hashing loss is better than standard triplet loss. Comparing PWCF-M with PWCF, the underlying manifold structure across different domains is extremely helpful to capture the correlation between samples. Comparing PWCF-H with PWCF, the proposed histogram features reduce the impact of data distribution discrepancy between different domains and it is unreasonable to use Euclidean distance of original features to measure the similarity between cross-domain samples.

4. Conclusion

In this paper, we propose an effective domain adaptive retrieval method named Probability Weighted Compact Feature Learning (PWCF), which learns compact binary codes to represent images. First, we propose BP induced focal-triplet loss, BP induced quantization loss and BP induced classification loss from the Bayesian perspective to optimize the binary compact feature between samples from different domains. Besides, The underlying manifold structure across different domains is used to capture meaningful nearest neighbors of different domains and further explore the potential correlation. To address the data distribution discrepancy issue, we propose a Histogram Feature of Neighbors (HFON) to metric the similarity/dissimilarity between the samples from different domains. The experimental results show that our PWCF always shows much higher retrieval performance in the scenarios of the cross-domain retrieval and single-domain retrieval, which verify that our method outperforms many state-of-the-art image retrieval methods.

References

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

[2] Mayur Datar, Piotr Indyk, Nicole Immorlica, and Vahab S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Twentieth Symposium on Computational Geometry*, 2004.

[3] Hal Daume III. Frustratingly easy domain adaptation. *ACL*, 2009.

[4] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: a deep convolutional activation feature for generic visual recognition. *International Conference on International Conference on Machine Learning*, 50(1):I–647, 2013.

[5] Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid. Mobile product image search by automatic query object extraction. *European Conference on Computer Vision*, 2012.

[6] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Similarity search in high dimensions via hashing. In *International Conference on Very Large Data Bases*, pages 518–529, 1999.

[7] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. 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, 2013.

[8] Y. Hao, T. Mu, J. Y. Goulermas, J. Jiang, R. Hong, and M. Wang. Unsupervised t-distributed video hashing and its deep hashing extension. *IEEE Transactions on Image Processing*, PP(99):1–1, 2017.

[9] Xiaofei He and Partha Niyogi. Locality preserving projections. In *Advances in neural information processing systems*, 2004.

[10] Junshi Huang, Rogerio S. Feris, Qiang Chen, and Shuicheng Yan. Cross-domain image retrieval with a dual attribute-aware ranking network. In *IEEE International Conference on Computer Vision*, 2015.

[11] Jonathan J. Hull. Database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 2002.

[12] Qing Yuan Jiang and Wu Jun Li. Scalable graph hashing with feature transformation. In *International Conference on Artificial Intelligence*, pages 2248–2254, 2015.

[13] Yu Gang Jiang, Jun Wang, Xiangyang Xue, and Shih Fu Chang. Query-adaptive image search with hash codes. *IEEE Transactions on Multimedia*, 15(2):442–453, 2013.

[14] Zhongming Jin, Cheng Li, Yue Lin, and Deng Cai. Density sensitive hashing. *IEEE transactions on cybernetics*, 44(8):1362–1371, 2014.

[15] Tang Jinhui, Li Zechao, Wang Meng, and Zhao Ruizhen. Neighborhood discriminant hashing for large-scale image retrieval. *IEEE Transactions on Image Processing*, 24(9):2827–2840, 2015.

[16] Weihao Kong and Wu Jun Li. Isotropic hashing. In *International Conference on Neural Information Processing Systems*, 2012.

[17] Yann Lecun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[18] Guosheng Lin, Chunhua Shen, Qinfeng Shi, Anton Van Den Hengel, and David Suter. Fast supervised hashing with decision trees for high-dimensional data. In *Computer Vision and Pattern Recognition*, pages 1971–1978, 2014.

[19] Hong Liu, Rongrong Ji, Jingdong Wang, and Chunhua Shen. Ordinal constraint binary coding for approximate nearest neighbor search. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, PP(99):1–1, 2018.

[20] J. Liu and L. Zhang. Optimal projection guided transfer hashing for image retrieval. In *AAAI Conference on Artificial Intelligence*, 2019.

[21] Wei Liu, Sanjiv Kumar, Sanjiv Kumar, and Shih Fu Chang. Discrete graph hashing. In *International Conference on Neural Information Processing Systems*, pages 3419–3427, 2014.

[22] Wei Liu, Jun Wang, Rongrong Ji, Yu-Gang Jiang, and Shih-Fu Chang. Supervised hashing with kernels. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2074–2081, 2012.

[23] Wei Liu, Jun Wang, Sanjiv Kumar, and Shih Fu Chang. Hashing with graphs. In *International Conference on Machine Learning, ICML 2011, Bellevue, Washington, Usa, June 28 - July*, pages 1–8, 2011.

[24] Xianglong Liu, Cheng Deng, Bo Lang, Dacheng Tao, and Xuelong Li. Query-adaptive reciprocal hash tables for nearest neighbor search. *IEEE Transactions on Image Processing*, 25(2):907–919, 2016.

[25] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S. Yu. Transfer feature learning with joint distribution adaptation. In *IEEE International Conference on Computer Vision*, 2014.

[26] Mohammad Norouzi, David J. Fleet, and Ruslan Salakhutdinov. Hamming distance metric learning. In *International Conference on Neural Information Processing Systems*, pages 1061–1069, 2012.

[27] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[28] Fumin Shen, Chunhua Shen, Wei Liu, and Heng Tao Shen. Supervised discrete hashing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 37–45, 2015.

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

[30] J. Tang, Z. Li, M. Wang, and R. Zhao. Neighborhood discriminant hashing for large-scale image retrieval. *IEEE Transactions on Image Processing*, 24(9):2827–2840, 2015.

[31] Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition*, pages 1521–1528, 2011.

[32] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *(IEEE) Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[33] Jun Wang, Wei Liu, A. X Sun, and Yu Gang Jiang. Learning hash codes with listwise supervision. In *IEEE International Conference on Computer Vision*, pages 3032–3039, 2013.

[34] Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. In *International Conference on Neural Information Processing Systems*, pages 1753–1760, 2008.

[35] Zaiwen Wen and Wotao Yin. A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 142(1-2):397–434, 2013.

[36] Peng Fei Zhang, Chuan Xiang Li, Meng Yuan Liu, Liqiang Nie, and Xin Shun Xu. Semi-relaxation supervised hashing for cross-modal retrieval. In *Acm on Multimedia Conference*, 2017.

[37] Joey Tianyi Zhou, Heng Zhao, Xi Peng, Meng Fang, Zheng Qin, and Rick Siow Mong Goh. Transfer hashing: From shallow to deep. *IEEE Transactions on Neural Networks and Learning Systems*, 2018.