TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 15/18 pp396–411
DOI: 10.26599/TST.2021.9010014
Volume 27, Number 2, April 2022

Asymmetric Deep Hashing for Person Re-Identifications

Yali Zhao, Yali Li, and Shengjin Wang*

Abstract: The person re-identification (re-ID) community has witnessed an explosion in the scale of data that it has to handle. On one hand, it is important for large-scale re-ID to provide constant or sublinear search time and dramatically reduce the storage cost for data points from the viewpoint of efficiency. On the other hand, the semantic affinity existing in the original space should be preserved because it greatly boosts the accuracy of re-ID. To this end, we use the deep hashing method, which utilizes the pairwise similarity and classification label to learn deep hash mapping functions, in order to provide discriminative representations. More importantly, considering the great advantage of asymmetric hashing over the existing symmetric one, we finally propose an asymmetric deep hashing (ADH) method for large-scale re-ID. Specifically, a two-stream asymmetric convolutional neural network is constructed to learn the similarity between image pairs. Another asymmetric pairwise loss is formulated to capture the similarity between the binary hashing codes and real-value representations derived from the deep hash mapping functions, so as to constrain the binary hash codes in the Hamming space to preserve the semantic structure existing in the original space. Then, the image labels are further explored to have a direct impact on the hash function learning through a classification loss. Furthermore, an efficient alternating algorithm is elaborately designed to jointly optimize the asymmetric deep hash functions and high-quality binary codes, by optimizing one parameter with the other parameters fixed. Experiments on the four benchmarks, i.e., DukeMTMC-reID, Market-1501, Market-1501+500k, and CUHK03 substantiate the competitive accuracy and superior efficiency of the proposed ADH over the compared state-of-the-art methods for large-scale re-ID.

Key words: person re-identification; deep hashing; asymmetric hashing; large-scale

1 Introduction

Person re-identification (re-ID) is usually treated as a retrieval problem: given a query based on a single image or a set of images, it aims to identify the matched identity from a large collection of gallery images captured from non-overlapping camera views by ranking these candidates according to some similarity metrics. In recent years, it has received increasing attention in practical intelligent vision systems.

Considering the explosive growing data for re-ID in the real world, it is quite necessary to efficiently search similar IDs in a given database for a query at the cost of acceptable storage and computational resource. Existing methods mainly focus on the effectiveness of re-ID, but the problem of efficiency is seriously understudied, despite its great importance. Hashing, a widely used technique for large-scale approximate nearest neighbor search, well meets the above requirements. It compresses high-dimensional data points into a Hamming space by generating compact codes and can simultaneously preserve the similarity and structural information of the original data, resulting in a balance of efficient computation and acceptable accuracy.

• Yali Zhao, Yali Li, and Shengjin Wang are with Department of Electronic Engineering, Tsinghua University, Beijing 100084, China. E-mail: zhaoluckydog@163.com; liyali@ocrserv.ee.tsinghua.edu.cn; wgsgj@tsinghua.edu.cn.

* To whom correspondence should be addressed.

Manuscript received: 2020-10-09; revised: 2021-02-16; accepted: 2021-02-23

Despite the great advantage of deep supervised hashing over the non-deep supervised one in many applications, most existing deep supervised hashing methods adopt a symmetric structure to learn one deep hash function. As defined in Ref. [1], given a pair of inputs x and x', symmetric hashing means that the similarity between them $S(x, x')$ is estimated by the Hamming distance between the outputs of the same hash mapping function, i.e., between $f(x)$ and $f(x')$, for some $f \in \{\pm\}^k$. The symmetric hashing is widely used but is inferior to the asymmetric one in many ways: it does not well explore the discriminative power of hashing method and is not efficient enough for training due to the difficulty of optimizing the discrete constraint. Moreover, the training of such symmetric hashing methods is typically time-consuming, making it hard to effectively utilize the supervised information for cases of large-scale databases.

Furthermore, as theoretically proven in Ref. [1], when approximating the similarity using the Hamming distance between binary hashes, shorter and more accurate hashes can be employed using two distinct hash mappings, i.e., by approximating the similarity between x and x' as the Hamming distance between $f(x)$ and $g(x')$ rather than between $f(x)$ and $f(x')$. Here f and g are two distinct hash functions to approximate the similarity affinity by the Hamming distance between two different binary hashing codes, which is called asymmetric hashing. Such asymmetry can preserve more semantic information thus resulting in a better retrieval accuracy, which works the same way even if the similarity metric is symmetric, e.g., even if it is based on the Hamming distance or Euclidean distance between asymmetric binary hashing codes.

In a word, compared with symmetric hashing, asymmetric hashing can obtain superior accuracy with shorter codes using two different hash mapping functions. Given the above analysis, we propose a concrete method called asymmetric deep ahshing (ADH) for re-ID, which is a convolutional neural network (CNN) based asymmetric hashing model for learning high-quality hashing codes. The overall view of the proposed method is illustrated in Fig. 1.

Specifically, a two-stream neural network is applied to construct a novel asymmetric structure, which directly learns two different hash mapping functions from pairs of images and takes advantage of the discriminative power of deep neural networks. The asymmetry of deep hash codes is further enhanced by preserving a better similarity between the binary codes of the training samples and the real-value features derived from the above hash mapping functions through an inner product.

We make use of the classification loss along with the above pairwise label to preserve more semantic information. In addition, the image labels have a direct impact on the hash mapping functions in our method, which is realized by constraining the learned binary hashing codes to be ideal for classification.

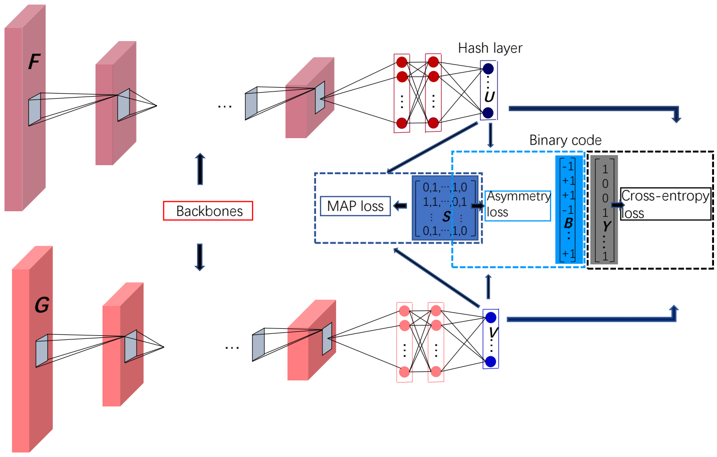


Fig. 1 Framework of the proposed ADH method. Two streams initialized with ImageNet pre-trained backbones are used for discriminative feature extraction, which are colored in red. Different shades of the red color represent the first stream *F* and second stream *G*, whose weights are asymmetrically different as a result of our alternating optimization strategy. The following two layers in blue denote the hashing layers as defined in Eq. (1), which is used to preserve the semantic similarity through the maximum a posteriori loss term, expressed in the dark blue-dotted frame. Furthermore, taking the binary hashing codes *B* into consideration, the explicit asymmetric loss between the real-value feature and the binary codes is proposed, expressed in the baby blue-dotted frame. Finally, a cross-entropy loss is introduced to exploit the point-wise semantic information, expressed in the black dotted frame.

The resulting problem is NP-hard, for which we propose an efficient alternating algorithm by optimizing over the binary hash codes and the hash functions in an iterative way. The three main contributions of our work are as follows:

- We propose a novel ADH method for large-scale re-ID. The asymmetry is reflected in two aspects: a two-stream deep neural network to asymmetrically learn two different hash mapping functions and an asymmetry term implemented with an inner product to reveal the similarity between the binary hash codes and the real-value representation derived from the hash functions. To the best of our knowledge, this is the first work to make use of the ADH for large-scale re-ID .
- An alternative minimization algorithm is designed to efficiently optimize the proposed formulation.
- Experiments show the effectiveness of the proposed ADH and its superiority over many existing state-of-the-art hashing methods on four large-scale re-ID datasets: DukeMTMC-reID, Market-1501, Market-1501+500k, and CUHK03. Combination and ablation studies are also performed to provide more insights into the proposed method.

The rest of this paper is organized as follows. A brief review of state-of-the-art hashing and re-ID methods is given in Section 2. The proposed ADH model for large-scale re-ID is presented in Section 3. Our experimental results are analyzed in Section 4. Finally, conclusions are drawn and further work is suggested in Section 5.

2 Related Work

This work focuses on a balance of the efficiency and effectiveness of large-scale re-ID through the ADH method. Thus, we mainly reviewed the typical works of re-ID, hashing methods, and hashing-based re-ID.

Person re-ID. Extensive research has been performed on re-ID, which mainly focuses on finding distinctive feature representation and learning discriminant models. A variety of distinctive visual features have been proposed to capture the appearance of a person under various conditions, e.g., color histogram[2], local binary pattern (LBP)[3], and local maximal occurrence (LOMO)[4]. Some typical metric learning techniques have also been used to discover a discriminative distance measure based on the learned feature, such as KISSME[5] and LMNN[6]. Many works have integrated the above components into an end-to-end framework. For instance, Varior et al.[7] incorporated long short-term memory modules into a Siamese network. Zhao et al.[8] proposed a deeply-learned part-aligned representation for re-ID, whereas Sun et al.[9] proposed a part-based convolutional baseline with a specific part pooling method.

All of the above methods achieved competitive performance, however, at the cost of large storage memory and efficient computation, which limits their application for large-scale re-ID in the real world.

Hashing methods. The hashing method has become a popular approach, to directly map an image to a binary code, resulting in a balance of effectiveness and efficiency. DSPH[10] was first proposed to utilize pairwise labels to train the end-to-end deep hashing model whereas SSDH[11] utilizes the Softmax classifier to train the hashing model. Deep semantic hashing (DSH)[12] was proposed to speed up the training of the network by adding a regular term instead of an activation function to the loss function. Moreover, DSH with generative adversarial network (GAN)[13] consists of a semi-supervised GAN to produce synthetic images, and a deep semantic hashing network with real-synthetic triplets to learn hash functions.

Deep hashing for re-ID. Given the trend of large-scale re-ID, several hashing based re-ID methods have been recently proposed. For example, Zhu et al.[14] proposed part-based deep hashing (PDH), which employs batches of triplet samples as the input of the deep hashing network. Wu et al.[15] proposed a structured loss function, which is defined over positive pairs and hard negatives to formulate a novel optimization problem. Zhu et al.[16] formulated the cross-view identity correlation and verification (X-ICE) hashing, which jointly learns cross-view identity representation binarization and discrimination in a unified manner.

3 Proposed Approach

In this section, we first present some notations used in this study and illustrate the problem to be solved. Then, we describe the proposed ADH for large-scale re-ID and design the alternative algorithm to efficiently optimize the proposed ADH.

3.1 Problem statement

The problem of large-scale re-ID aims to identify a set of n_g gallery (target) people $\tilde{G} = \{\tilde{I}_i^g\}_{i=1}^{n_g}$ captured from m_g cameras $\{\mathrm{Cam}_i^g\}_{i=1}^{m_g}$ in deployment. The search space contains n_p probe person images $\tilde{P} = \{\tilde{I}_i^p\}_{i=1}^{n_p}$ captured by m_p cameras $\{\mathrm{Cam}_i^p\}_{i=1}^{m_p}$ without an overlap against any of the n_g gallery cameras, i.e., cross-view re-ID matching.

The training set contains n_t training people $T = \{I_i^t\}_{i=1}^{n_t}$ captured from m_t cameras $\{\mathrm{Cam}_i^t\}_{i=1}^{m_t}$. The labels of the n_t training samples are donated as $Y = [y_1, y_2, \ldots, y_{n_t}]$, where $y_i \in \{0,1\}^{c\times 1}$ corresponds to the training sample I_i^t and c is the number of categories in the training set. Furthermore, the pairwise label information can be derived as $S = \{s_{ij}\}, s_{ij} \in \{0,1\}$, where $s_{ij} = 1$ means that the samples I_i^t and I_j^t are semantically similar and $s_{ij} = 0$ means that they are semantically dissimilar.

For model formulation, we aim to learn hash mapping functions $\mathbb{R}^d \rightarrow \{-1,1\}^K$ from the training set that can be applied to convert a test image into short K-dim hash codes followed by a fast search with the Hamming distance.

3.2 ADH for person Re-ID

We use the uppercase letters $T = [I_{1F}^t, I_{2F}^t, \ldots, I_{n_t F}^t] \in \mathbb{R}^{n_t \times d_1 \times d_2 \times 3}$ and $T = [I_{1G}^t, I_{2G}^t, \ldots, I_{n_t G}^t] \in \mathbb{R}^{n_t \times d_1 \times d_2 \times 3}$ to denote the input images in the first stream F and second stream G, respectively. d_1 and d_2 are the width and length of a training image, respectively. Here we denote the samples in F and G with the same symbol T, which means that the n_t training samples are alternatively used in the first and second networks. The parameters of these two hash mapping functions F and G are denoted as θ_F and θ_G, respectively. The binary codes of the whole training set are denoted as $B = [b_1, b_2, \ldots, b_{n_t}]^{\mathrm{T}}$, where the K-bit binary code of the i-th sample is $b_i \in \{-1,+1\}^{K\times 1}$. Overall, we propose such an asymmetric model to learn two different hash functions F and G and one consistent binary code b_i for each image at the training phase.

Given a training sample I_i^t, the binary representations from the two streams are denoted as $u_i = F(I_i^t, \theta_F) \in \{-1,+1\}^{K\times 1}$, and $v_i = G(I_i^t, \theta_G) \in \{-1,+1\}^{K\times 1}$. The overall binary representations of the whole training set are accordingly denoted as $U = [u_1, u_2, \ldots, u_{n_t}]^{\mathrm{T}} \in \mathbb{R}^{n_t \times K}$ and $V = [v_1, v_2, \ldots, v_{n_t}]^{\mathrm{T}} \in \mathbb{R}^{n_t \times K}$.

The proposed ADH for large-scale re-ID is based on a two-stream framework and consists of four components: (1) a two-stream subnetwork to capture discriminative real-value representations. Without loss of generality, ResNet-50[17] is used as the backbone and features derived from conv5 are extracted as the final real-value representations; (2) an additional hashing layer H to generate compact binary hash codes from the above real-value data; (3) an asymmetric loss for similarity-preserving learning between the real-value representations and the binary hash codes; and (4) a cross-entropy loss designed for semantic hashing codes. The detailed configuration of our ADH model is shown in Fig. 1.

The additional hashing layer H, which is designed to transform the real-value data from conv5 into K-dimension binary codes U and V, is implemented with an additional fully-connected (FC) layer with K hidden units. For a given pair of training samples I_i^t and I_j^t with real-value feature vectors $a_i^5 \in \mathbb{R}^d$ and $a_j^5 \in \mathbb{R}^d$ from streams F and G, the corresponding K-dimensional binary hashing representations can be computed as follows:

$$\begin{aligned} f_i^H &= \mathrm{sgn}(a_i^5 W_F^H + b_F^H) = \mathrm{sgn}(f_i), \\ g_j^H &= \mathrm{sgn}(a_j^5 W_G^H + b_G^H) = \mathrm{sgn}(g_j) \end{aligned} \tag{1}$$

where $\mathrm{sgn}(x) = 1$ if $x > 0$ and -1 otherwise, and $\mathrm{sgn}(\cdot)$ performs element-wise operations for a matrix or a vector. W is the weight between conv5 and the additional hashing layer.

Considering it is difficult to make a backpropagation for the gradient with respect to $\mathrm{sgn}(\cdot)$ because their gradients are zero anywhere, we adopted $\tanh(\cdot)$ to softly approximate the $\mathrm{sgn}(\cdot)$ function, that is, $u_i = \tanh(f_i)$ and $v_j = \tanh(g_j)$. Given such a formulation, the following loss functions were designed to derive well-performed hashing codes.

3.2.1 Maximum a posteriori loss

We propose to preserve the pairwise similarity by controlling the quantization error in a Bayesian framework. Given the pairwise similarity $S = \{s_{ij}\}$ and the binary hashing codes of the training samples U and V, the maximum a posterior (MAP) estimation of the hash codes can be represented as

$$p(U,V|S) \propto p(S|U,V)p(U,V) = \sum_{s_{ij}\in S} p(s_{ij}|u_i,v_j)p(u_i,v_j) \tag{2}$$

where $p(S|U,V)$ denotes the likelihood function and $p(U,V)$ is the prior distribution. For each pair of images I_i^t and I_j^t, $p(s_{ij}|u_i,v_j)$ is the conditional probability of the similarity information s_{ij} given their binary hashing codes u_i and v_j, which can be formulated as

$$p(s_{ij}|U,V) = \begin{cases} \sigma(\phi_{ij}), & s_{ij} = 1; \\ 1 - \sigma(\phi_{ij}), & s_{ij} = 0 \end{cases} \tag{3}$$

where $\sigma(\cdot)$ is the sigmoid function with the formulation $\sigma(x) = 1/(1+\mathrm{e}^{-x})$. Here ϕ_{ij} can be defined as $\phi_{ij} = 1/2\langle u_i, v_j\rangle = 1/2u_i^{\mathrm{T}}v_j$, because it holds that $dist_H(u_i,v_j) = 1/2(K - \langle u_i, v_j\rangle)$.

The following negative log likelihood function can be used as the loss function to make the Hamming distance of two similar samples as small as possible and simultaneously make that of the dissimilar ones as large as possible.

$$l_{MAP} = -\log p(S|U,V) = -\sum_{s_{ij}\in S} \log p(s_{ij}|U,V) = \\ -\sum_{s_{ij}\in S} (s_{ij}\phi_{ij} - \log(1+\mathrm{e}^{\phi_{ij}})) \tag{4}$$

Remarks. Accordingly, a novel ADH structure is proposed. The two streams of CNN were trained to asymmetrically learn two different hash mapping functions as shown in the following optimization algorithm.

3.2.2 Asymmetry loss

The l_{MAP} loss realizes the asymmetry of hash codes to some extent, reflected in the alternative optimization algorithm of the two CNN streams. In this section, we propose to further enhance the asymmetry to make full use of its advantage. Specifically, we propose to preserve the similarity between the real-value representations produced by the hash mapping functions and the directly learned binary hash codes B. To this end, another asymmetric pairwise loss is formulated as follows:

$$l_{Asy} = \sum_{s_{ij}} \|\tanh(f_i^{\mathrm{T}})b_j - Ks_{ij}\|_2^2 + \\ \|\tanh(g_i^{\mathrm{T}})b_j - Ks_{ij}\|_2^2 = \\ \sum_{s_{ij}} \|u_i^{\mathrm{T}}b_j - Ks_{ij}\|_2^2 + \|v_i^{\mathrm{T}}b_j - Ks_{ij}\|_2^2, \\ \text{s.t. } b_j \in \{-1,+1\}^{K\times 1} \tag{5}$$

As can be seen, the similarity between the binary hash codes b_j and real-value representation $f_i(g_i)$ derived from the learned hash mapping functions $F(G)$ is measured by their inner product. Such a formulation not only encourages $\tanh(f_i)$ ($\tanh(g_i)$) and b_j to be consistent, but also preserves the similarity between them through the supervised information s_{ij}. Furthermore, considering the fact that the binary code of the training sample I_i^t is derived from the learned hash mapping functions F and G in the form of u_i and v_i, the binary code of the training sample I_j^t is produced by the directly learned hash code B, which is totally an explicit definition of asymmetric hashing. Thus the proposed l_{Asy} loss enhances the asymmetry of the model and further takes advantage of its great effectiveness and efficient training as theoretically proven in Ref. [1].

Remarks. The asymmetry of deep hashing is enhanced by revealing the similarity between the real-value features and binary codes through an additional asymmetric loss, which provides a much more discriminative hashing code.

3.2.3 Cross-entropy loss

We formulated an additional classification layer to make full use of the semantic labels in a point-wise way. Most of the previous works[18, 19] made use of the label information under a two-branch multitask learning framework, i.e., a classification stream to measure the classification error and another separate hashing stream to learn the hash function, where the classification stream is only employed to learn the image representations and has no direct impact on the hash mapping functions. Instead, we make full use of the label information by directly constraining the learned binary hashing codes to be ideal for the classification as follows:

$$Y = W_F^{\mathrm{T}}U, \quad Y = W_G^{\mathrm{T}}V \tag{6}$$

where $W_F = [w_F^1, w_F^2, \ldots, w_F^K]$ and $W_G = [w_G^1, w_G^2, \ldots, w_G^K]$ are the classifier weights, implemented with an additional FC layer following the hashing layer. Then the cross-entropy classification loss can be calculated with l_2 loss as follows:

$$l_{CE} = \sum_{i=1}^{n_t}(\|y_i - W_F^{\mathrm{T}}u_i\|_2^2 + \|y_i - W_G^{\mathrm{T}}v_i\|_2^2) \tag{7}$$

Remarks. Such a l_{CE} loss makes the learned binary codes optimal for the learned linear classifier, resulting in more semantic hashing codes.

3.3 Overall objective

Combining all the analysis results presented above, we obtain the overall objective loss function as follows:

$$l_{all} = l_{MAP} + \alpha l_{Asy} + \beta l_{CE} = -\sum_{s_{ij}\in S}[(s_{ij}\phi_{ij} - \log(1+\mathrm{e}^{\phi_{ij}})) + \\ \alpha(\|u_i^{\mathrm{T}}b_j - Ks_{ij}\|_2^2 + \|v_i^{\mathrm{T}}b_j - Ks_{ij}\|_2^2)] + \\ \beta\Big(\sum_{i=1}^{n_t}(\|y_i - W_F^{\mathrm{T}}u_i\|_2^2 + \|y_i - W_G^{\mathrm{T}}v_i\|_2^2)\Big) \tag{8}$$

where α and β are the non-negative parameters to make a trade-off among the three losses.

3.4 Optimization

As shown in Eq. (8), the parameters to be optimized consist of the weights F (G) from the backbones and additional hashing layers, the weights of the linear classifiers W_F (W_G), and the discrete binary codes B. Generally, the proposed model Eq. (8) is a mixed-integer programming problem, which is non-convex and nonsmooth, thus resulting in an NP-hard problem due to the binary constraint B. To address this problem, a well-designed alternative optimization algorithm was exploited, where each subproblem can be efficiently solved, yielding satisfactory final solutions. That is, only one variable was optimized with the others fixed for each subproblem. Specifically, considering the asymmetry of the proposed method, we sequentially updated the parameters of the two streams and binary code matrix B in an alternative manner.

3.4.1 Initialization

We made use of the ImageNet pre-trained model to initialize the two backbones and the principled initialization methods (e.g., k-means clustering) for W_F and W_G as we empirically found that it performs better than random initialization. Moreover, we initialized B by randomly setting $l(l < K)$ entries in each column b of B to 1.

3.4.2 ($\boldsymbol{\theta_F}$,$\boldsymbol{W_F}$)-step

For this step, we update the parameters (θ_F, W_F) of the first stream with (θ_G, W_G) of the second stream and the binary code B fixed. Thus, the objective function Eq. (8) can be simplified as

$$l_{all} = -\sum_{s_{ij}\in S}[(s_{ij}\phi_{ij} - \log(1 + \mathrm{e}^{\phi_{ij}}))+$$

$$\alpha(\|u_i^{\mathrm{T}}b_j - Ks_{ij}\|_2^2)] + \beta\Big(\sum_{i=1}^{n_t}(\|y_i - W_F^{\mathrm{T}}u_i\|_2^2\Big) \quad (9)$$

Then we aim at updating (θ_F, W_F) with mini-batch stochastic gradient descent (SGD) backpropagation. For simplicity, we donated the image label $W_F^{\mathrm{T}}u_i$ produced by the linear classifier as $\hat{y}_i$. Thus, the gradient of the objective function with respect to $\hat{y}_i$ is

$$\frac{\partial l_{all}}{\partial \hat{y}_i} = -2\beta(y_i - W_F^{\mathrm{T}}u_i) \quad (10)$$

According to the chain rule, the gradient of the parameters W_F in the first linear classifier can be given as

$$\frac{\partial l_{all}}{\partial W_F} = \sum_{i=1}^{n_t}\frac{\partial l_{all}}{\partial \hat{y}_i}\frac{\partial \hat{y}_i}{\partial W_F} = -2\beta\Big(\sum_{i=1}^{n_t}(y_i - W_F^{\mathrm{T}}u_i)u_i\Big) \quad (11)$$

The gradient with respect to the real-value f_i can be derived as

$$\frac{\partial l_{all}}{\partial f_i} = \alpha\sum_{j=1}^{n_t}\Big[2b_j(b_j^{\mathrm{T}}u_i - Ks_{ij}) + \frac{1}{2}(\theta(\phi_{ij})v_j - s_{ij}v_j)\Big]\odot$$

$$(1 - u_i^2) - 2\beta\Big(\sum_{i=1}^{n_t}(y_i - W_F^{\mathrm{T}}u_i)u_i\Big) \quad (12)$$

where $\odot$ denotes the dot product. After deriving the gradient $\frac{\partial l_{all}}{\partial f_i}$, the chain rule was used to obtain $\frac{\partial l_{all}}{\partial \theta_F}$, and θ_F was updated using backpropagation. All in all, the parameters (θ_F, W_F) of the first stream were updated.

3.4.3 ($\boldsymbol{\theta_G}$,$\boldsymbol{W_G}$)-step

Similarly, we updated the parameters (θ_G, W_G) of the second stream through backpropagation with (θ_F, W_F) of the first stream and binary code B fixed. The objective function Eq. (8) can be simplified as

$$l_{all} = -\sum_{s_{ij}\in S}[(s_{ij}\phi_{ij} - \log(1 + \mathrm{e}^{\phi_{ij}}))+$$

$$\alpha(\|v_i^{\mathrm{T}}b_j - Ks_{ij}\|_2^2)] + \beta\Big(\sum_{i=1}^{n_t}(\|y_i - W_G^{\mathrm{T}}v_i\|_2^2\Big) \quad (13)$$

The gradient of the objective function with respect to $\hat{y}_i$ and further to W_G can be derived respectively as follows:

$$\frac{\partial l_{all}}{\partial \hat{y}_i} = -2\beta(y_i - W_G^{\mathrm{T}}v_i),$$

$$\frac{\partial l_{all}}{\partial W_G} = \sum_{i=1}^{n_t}\Big(\frac{\partial l_{all}}{\partial \hat{y}_i}\frac{\partial \hat{y}_i}{\partial W_G}\Big) = -2\beta\Big(\sum_{i=1}^{n_t}(y_i - W_G^{\mathrm{T}}v_i)v_i\Big) \quad (14)$$

Thus, the gradient with respect to the real-value g_i can be derived as

$$\frac{\partial l_{all}}{\partial g_i} = \alpha\sum_{j=1}^{n_t}[2b_j(b_j^{\mathrm{T}}v_i - Ks_{ij}) + \frac{1}{2}(\theta(\phi_{ij})u_j - s_{ij}u_j)]\odot$$

$$(1 - v_i^2) - 2\beta\Big(\sum_{i=1}^{n_t}(y_i - W_G^{\mathrm{T}}v_i)v_i\Big) \quad (15)$$

After deriving the gradient $\frac{\partial l_{all}}{\partial g_i}$, the chain rule was used to obtain $\frac{\partial l_{all}}{\partial \theta_G}$, and θ_G is updated using backpropagation. Up until now, the parameters (θ_G, W_G) of the first stream are still being updated.

3.4.4 *B*-step

For this step, we update the binary code B with (θ_F, W_F) of the first stream and (θ_G, W_G) of the second stream fixed. The objective function Eq. (8) can be simplified as

$$l_{all} = \sum_{s_{ij}\in S}\alpha(\|u_i^{\mathrm{T}}b_j - Ks_{ij}\|_2^2 + \|v_i^{\mathrm{T}}b_j - Ks_{ij}\|_2^2) =$$

$$\alpha(\|UB^{\mathrm{T}} - KS\|_2^2 + \|UB^{\mathrm{T}} - KS\|_2^2) =$$

$$-2\alpha Tr(B(K(U^{\mathrm{T}}S + V^{\mathrm{T}}S)))+$$

$$\|BU^{\mathrm{T}}\|_2^2 + \|BV^{\mathrm{T}}\|_2^2 + const,$$

$$\text{s.t. } b_j \in \{-1, +1\}^{K\times 1} \tag{16}$$

where *const* is a constant value without any association with B. In this study, we used the discrete cyclic coordinate descend method to iteratively solve B column by column as follows. For simplicity, we donated $-2\alpha(K(S^{\mathrm{T}}U + S^{\mathrm{T}}V))$ as Q, so Formula (16) can be simplified as

$$l_{all} = Tr[BQ^{\mathrm{T}}] + \|BU^{\mathrm{T}}\|_2^2 + \|BV^{\mathrm{T}}\|_2^2 + const,$$
$$\text{s.t. } b_j \in \{-1, +1\}^{K\times 1} \tag{17}$$

where *Tr* is the trace of a matrix.

Furthermore, we donated B_{*c} as the c-th column and $\hat{B}_c$ as the left columns of B. The same formulation applied to U_{*c}, $\hat{U}_c$, V_{*c}, and $\hat{V}_c$. Thus, Formula (17) can be formulated as

$$l_{all} = Tr(B_{*c}[2(U_{*c}^{\mathrm{T}}\hat{U}_c + V_{*c}^{\mathrm{T}}\hat{V}_c)\hat{B}_c^{\mathrm{T}} + Q_{*c}^{\mathrm{T}}]) + const,$$
$$\text{s.t. } b_j \in \{-1, +1\}^{K\times 1} \tag{18}$$

The optimal solution for B_{*c} can be gained as

$$B_{*c} = -\mathrm{sgn}(2\hat{B}_c(\hat{U}_c^{\mathrm{T}}U_{*c} + \hat{V}_c^{\mathrm{T}}V_{*c}) + Q_{*c}) \tag{19}$$

After computing B_{*c}, B can be updated by replacing the c-th column with B_{*c} and Eq. (19) will be repeated until all columns are updated. This method is called the discrete cyclic coordinate descend method.

The overall learning algorithm of the proposed ADH is briefly summarized in Algorithm 1.

3.5 Testing

In the testing phase, the learned hash functions F and G can be applied to generate binary codes for query and gallery images in the testing stage. Specifically, for a testing image I_i, its binary hashing codes from the two streams are $b_F = \mathrm{sgn}(F(I_i, \theta_F))$ and $b_G = \mathrm{sgn}(G(I_i, \theta_G))$ computed in Eq. (1), respectively.

We experimentally found that compared with a single stream, i.e., the first or second stream, the averaged outputs of the two stream networks can provide a more robust and discriminative representation of an input, so we generated the binary hashing code for I_i in the testing stage as follows:

Algorithm 1 Learning algorithm for ADH

```
Input: Training images $T = \{I_i^t\}_{i=1}^{n_t}$, their corresponding
    labels $Y$ and similarity matrix $S = \{s_{ij}\}, s_{ij} \in \{0, 1\}$, the
    hashing code length $K$, and the hyper parameters $\alpha$ and $\beta$.
Output: The weights of the first stream $(\theta_F, W_F)$ and the
    second stream $(\theta_G, W_G)$, and the discrete binary codes $B$.
    Initialization: Initialize $\theta_F$ and $\theta_G$ with pre-trained ResNet-
    50, $W_F$ and $W_G$ with the principled initialization methods, and
    $B$ by randomly setting $l$ ($l < K$) entries in each column of $B$
    to 1.
    while not reach the maximum iteration
    do
    1. $(\theta_F, W_F)$-step: Update $(\theta_F, W_F)$ with $(\theta_G, W_G)$ and $B$
    fixed according to Eq. (12)
    2. $(\theta_G, W_G)$-step: Update $(\theta_G, W_G)$ with $(\theta_F, W_F)$ and $B$
    fixed according to Eq. (15)
    3. $B$-step: Update $B$ with $(\theta_F, W_F)$ and $(\theta_G, W_G)$ fixed
    according to Eq. (19)
    end while
```

$$b_i = \mathrm{sgn}\left(\frac{1}{2}(F(I_i, \theta_F) + G(I_i, \theta_G))\right) \tag{20}$$

4 Experiment

4.1 Datasets and evaluation protocol

Datasets. We evaluated the performance of proposed method on four widely used largest re-ID datasets: DukeMTMC-reID[20], Market-01[21], Market-1501+500k[21], and CUHK03[22], some samples of whose are shown in Fig. 2. These datasets were chosen due to their large scales, for which effective retrieval methods are of great advantage.

- The DukeMTMC-reID dataset is one of the most challenging re-ID datasets up to now. It contains 1404 identities, 16 522 training images, 2228 queries, and 17 661 gallery images.
- Market-1501 is composed of 19 732 gallery images, 3368 query images, and 1501 identities automatically detected from six cameras. The training set contains 12 936 images of 751 identities. The testing set has 19 732 images of 750 identities.
- The distractor set Market-1501+500k contains

(a) Market-1501 (b) Market-1501+500k (c) DukeMTMC-reID (d) CUHK03

Fig. 2 Example of person images, with two images in each column corresponding to the same person for every dataset.

5×10^5 images which are treated as outliers besides the 32 668 bounding boxes of 1501 identities.

- CUHK03 consists of 1467 identities from 6 camera views deployed on a university campus. This dataset was constructed by manual labeling and autodetection (DPM). We report our results under a hard setting called CUHK03-NP, resently proposed in Ref. [23], where 767 identities are used for training and 700 identities for testing.

For simplicity and focus on the proposed algorithm itself, all the following experiments were evaluated under the single query setting and without the use of a re-ranking[23] algorithm. Particularly, the multi-query and re-ranking strategies can greatly boost the final mean average precision (mAP).

Evaluation protocol. We adopted the widely used cumulated matching characteristics (CMC) approach[24] for the quantitative evaluation, a standard evaluation metric in re-ID. The top-n matching rate indicates the expectation of finding anyone of the correct matched images and rank-1 is widely used, which is abbreviated as R-1. As in Ref. [21], where re-ID was mainly treated as a retrieval problem, the mAP was also used in our experiments to evaluate the performance. It is calculated as the mean score of the average precision (AP) of all the query images. AP is calculated by the area under the precision-recall curve.

4.2 Implementation details

Model training. The mini-batch SGD was adopted with a momentum of 0.9 and weight decay of 0.0005. Because the weights θ_F and θ_G are initialized with pre-trained ImageNet model while the linear classifier W_F and W_G were trained from scratch far away from the final solution, we set the learning rates of W_F and W_G to be ten times those of the lower layers.

We adopted the warmup learning rate policy[25], which contains two stages, i.e., fine-to-coarse and coarse-to-fine stages:

$$lr(t) = \begin{cases} 3 \times 10^{-5} \times \dfrac{t}{30}, & t \leqslant 30; \\ 3 \times 10^{-4}, & 30 \leqslant t < 80; \\ 3 \times 10^{-5}, & 80 \leqslant t < 120; \\ 3 \times 10^{-6}, & 120 \leqslant t < 150. \end{cases}$$

Training strategy. In learning models that correspond to different hashing code lengths, training each model from scratch would be a severely computational waste because the knowledge distilled by the preceding layers can be shared by these models. Moreover, as the code length grows, the model would contain more parameters in the output layer and thus becomes prone to overfitting. Given these considerations, we particularly designed the training methodology as follows: first, train the model with a few nodes in the output layer, and then fine-tune it to derive the target model with the desired code length.

Mini-batch construction. As for the generation of image pairs, we exploited all the unique pairs in each mini-batch online so that we can make better use of computational resources and storage space, making it possible to scale up to large-scale datasets. For each mini-batch, we randomly selected P different persons without replacement, and for each person N images were randomly chosen. Thus, there are a total of $P \times N$ images in a mini-batch. An epoch means that all persons are sampled. During training, the batch size was 128 with $P = 32$ and $N = 4$.

4.3 Performance evaluation

We conducted extensive experiments to evaluate the effectiveness and efficacy of the proposed ADH against several state-of-the-art hashing or non-hashing methods on four widely-used large-scale benchmarks.

4.3.1 Comparison with the state-of-the-art hashing methods

To demonstrate the effectiveness of the proposed ADH, we made a wide comparison with existing hashing methods: hashing methods designed for general purpose and those particularly designed for re-ID.

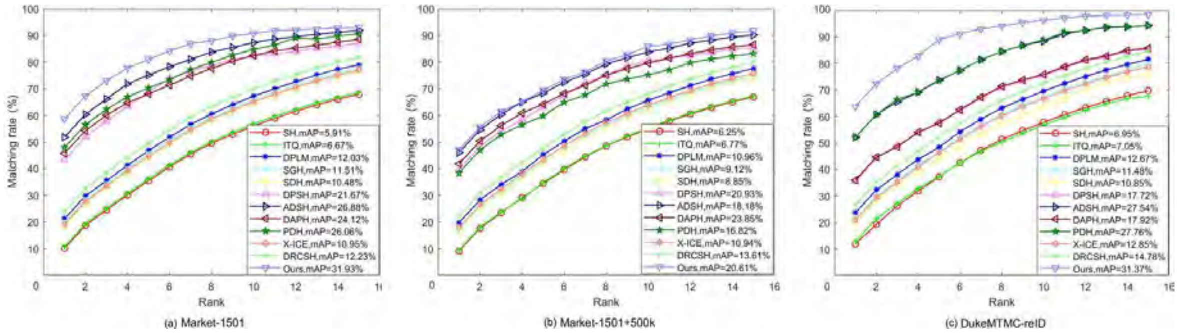
The first category consists of the traditional hashing methods and deep hashing methods, where the traditional ones can be further divided into unsupervised and supervised methods. Specifically, the unsupervised traditional hashing methods include SH[26] and ITQ[27]. The compared supervised traditional hashing methods include DPLM[28], SGH[29], and SDH[30], where the features extracted by the backbone were used as the input of the hashing methods. The compared deep learning based hashing methods include DPSH[10], ADSH[31], DAPH[32], and Greedy Hash[33]. The second category consists of PDH[14], X-ICE[16], and DRCSH[34].

The results on the datasets are summarized in Table 1 and Fig. 3. The following is a detailed explanation of our findings.

- Compared with unsupervised hashing methods, conventional non-deep supervised hashing methods generally achieve better performance owing to their wealth of supervised information.
- The deep hashing methods can outperform the nondeep ones, benefiting from the discriminative deep

Table 1 Performance comparison with state-of-the-art hashing methods. The four partitionings are the unsupervised traditional hashing methods, supervised traditional hashing methods, deep learning based hashing methods, and hashing methods particularly designed for re-ID.

Method	Market-1501		Market-1501+500k		DukeMTMC-reID		CUHK03-NP(detected)		CUHK03-NP(labeled)	
	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP
SH[26]	10.15	5.91	9.12	6.25	11.93	6.95	5.45	5.39	5.91	9.85
ITQ[27]	10.72	6.67	9.64	6.77	12.89	7.05	5.67	5.51	6.23	6.11
DPLM[28]	21.22	12.03	19.78	10.96	23.74	12.67	12.03	11.97	12.31	12.04
SGH[29]	19.92	11.51	17.15	9.12	21.46	11.48	11.56	11.23	11.91	11.34
SDH[30]	18.76	10.48	16.73	8.85	19.95	10.85	11.23	11.05	11.67	11.15
DPSH[10]	43.33	21.67	39.75	20.93	35.61	17.72	28.12	27.91	28.36	28.05
ADSH[31]	51.78	26.88	45.95	18.18	52.13	27.54	31.65	31.12	32.03	31.81
DAPH[32]	45.67	24.12	41.78	23.85	36.03	17.92	29.27	28.83	29.39	29.01
Greedy Hash[33]	49.56	26.98	45.73	26.22	40.56	23.76	34.67	34.12	34.78	34.34
PDH[14]	47.89	26.06	38.39	16.82	52.04	27.76	31.17	29.45	32.15	30.12
X-ICE[16]	19.27	10.95	17.97	10.94	20.92	12.85	15.21	15.11	15.72	15.35
DRCSH[34]	24.08	12.23	22.25	13.61	26.77	14.78	17.78	17.23	18.05	17.42
Ours	58.67	31.93	47.03	20.61	63.76	31.37	33.21	30.13	35.12	31.97



(a) Market-1501 (b) Market-1501+500k (c) DukeMTMC-reID

Fig. 3 CMC curves of the state-of-the-art hashing methods on Market-1501, Market-1501+500k, and DukeMTMC-reID datasets.

binary codes and the interaction between the binary code inference step and hash mapping function learning step based on the learned codes in an end-to-end manner.

- In most cases, re-ID dedicated methods can outperform those intended for general purposes. This finding is reasonable because these task-dedicated methods provide re-ID-specific domain knowledge and insights. For example, in Ref. [16], the significant appearance variation of a person was dealt with due to view transformation, pose and lighting condition change, and occlusion among others, specifically under the cross-view search setting. Thus, considerable fine-grained information derived from the re-ID problem can be mined to boost the final performance.

- Our method can substantially outperform all the compared methods. For instance, with respect to the mAP, compared with the PDH, which is the second-best method, the proposed ADH shows superior performance gains of 5.87%, 3.79%, 3.61%, 0.7%, and 1.85% on the compared datasets, respectively.

We mainly contribute these findings to our wellformulated loss functions, especially the asymmetric ones, either implicitly through the hash mapping functions or explicitly through the constraint in the feature space. Moreover, the unified framework, which simultaneously learns the discriminative image representation and hash coding, plays a vital role in good performance.

- Compared with the asymmetric methods ADSH[31] and DAPH[32], our proposed method exceeds the best one by a much larger margin. The underlying reason may be that by alternatively training two streams of deep networks to construct distinct asymmetric hash mapping functions, our method has higher learning capability and thus can capture more information than a symmetric hashing setting. Besides, the explicit asymmetry constraint on the real-value data and the binary hash code B can exploit more semantic information in the feature space, further boosting its advantage over the symmetric ones.

4.3.2 Comparison with state-of-the-art re-ID methods

We compared the proposed ADH with the state-of-the-art re-ID methods, which are categorized into two groups, i.e., the two-step method consisting of a feature extraction step and a metric learning step, either based on handcrafted features or deep features from ResNet-50[17], and the one-stage methods, i.e., deep learning methods.

We consider 13 typical methods. Specifically, for the first category, the compared methods of the handcrafted feature + metric learning category consists of BoW + KISSME[29] and LOMO + XQDA[4], and the deep feature + metric learning category consists of deep feature + WARCA[35] and deep feature + KLFDA[36]. For the second category, we made a comparison with SVDNet[9], PCB+RPP[37], MultiScale[38], EdgeNet[39], DropEasy[40], MuDeep (SL)[41], GLAD[42], A^3M[43], and P^2-*net*[44].

The re-ID performance of the four datasets is detailed in Table 2, and the following conclusions can be drawn:

- ADH brings great improvement over the traditional pipelines of handcrafted features followed by a metric learning step. For example, ADH exceeds the LOMO + XQDA[4] by a large margin of 9.73% and 14.13% in mAP and 14.25% and 33.01% in Rank-1 on the Market-1501 and DukeMTMC-reID, respectively, which demonstrates the effectiveness of the proposed ADH method for largescale re-ID, benefiting from the deep features and the one-step end-to-end pipeline.
- Compared with the deep features extracted by ResNet-50[17], followed by a metric learning step, the proposed ADH is very competitive. For example, the mAPs of the compared datasets for deep feature + WARCA[35] are 9.81%, 6.63%, 2.62%, 14.3%, and 14.65% lower than our method, which verifies the effectiveness of simultaneously learning the binary codes and hash mapping functions.
- All of the compared models based on deep methods nearly achieved a competitive performance over ours, but at the expense of the high computation and storage efficiency, which limits their application for large-scale re-ID in the real world. The performance gap of our method is partially attributed to information loss in converting long real-value representations into short binary hash codes. The advantage of ADH lies in the critical computational and storage efficiency over all these re-ID competitors, which enables fast re-ID in large-scale galleries.

4.3.3 Ablation study

In this section, we performed an ablation study of each component of the network in two aspects: (1) the effect of different loss terms and (2) the effect of different backbones.

Impact of different loss terms. We introduce four stripped-down variants of the full ADH in Eq. (8) to investigate the impact of different loss terms on the final re-ID performance.

- **ADH/B:** This is a variant of ADH without the binarization operation on the discriminative real-value representation, as defined in Eq. (1), which may serve as an upper bound of the overall performance.
- **ADH/Asy:** We set $\alpha = 0$, which implies a lack of an explicit asymmetric loss term from Eq. (8). This step

Table 2 Performance comparison with state-of-the-art re-ID methods. DF: deep feature.

Method	Market-1501		Market-1501+500k		DukeMTMC-reID		CUHK03-NP(detected)		CUHK03-NP(labeled)	
	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP
BoW + XQDA[29]	44.42	20.76	13.85	–	25.13	12.17	6.40	6.40	7.90	7.31
LOMO + XQDA[4]	43.8	22.2	–	–	30.75	17.04	12.80	11.50	14.80	13.60
DF + WARCA[35]	45.2	22.12	27.67	13.98	50.21	28.75	23.45	15.83	25.67	17.23
DF + KLFDA[36]	51.4	24.4	33.67	15.79	55.69	30.34	24.57	16.73	25.61	17.12
SVDNet[9]	82.3	62.1	–	–	76.7	56.8	41.50	37.30	–	–
PCB+RPP[37]	93.8	81.6	72.59	70.22	83.3	69.2	63.70	57.50	–	–
MultiScale[38]	88.9	73.1	70.36	65.87	79.2	60.6	43.0	40.5	40.7	37.0
EdgeNet[39]	80.20	–	–	–	–	–	–	–	–	–
DropEasy[40]	93.8	78.3	–	–	58.9	78.6	40.0	46.4	55.9	50.4
MuDeep (SL)[41]	95.34	84.66	–	–	88.19	75.63	71.93	67.21	75.64	70.54
GLAD[42]	89.9	73.9	–	–	80.0	62.2	83.3	–	86.0	–
A^3M[43]	86.54	68.97	–	–	–	–	–	–	–	–
P^2-*net*[44]	95.2	85.6	–	–	86.5	73.1	74.9	68.9	78.3	73.6
Ours	58.67	31.93	47.03	20.61	63.76	31.37	33.21	30.13	35.12	31.97

allows evaluating the effectiveness of a combination of implicit asymmetry of hash mapping functions as a result of the MAP estimation term and point-wise semantics of the hashing codes.

- **ADH/CE-v1:** We set $\beta = 0$, which implies removing the point-wise semantic constraint of the hashing codes from the overall objective (Eq. (8)). This step allows evaluating the effectiveness of the asymmetry of the hashing codes, either implicitly with the help of MAP loss term or explicitly through the asymmetric loss term.
- **ADH/CE-v2:** For this variant of ADH, the cross-entropy loss is applied to the real-value features derived from the backbones, that is, the layer before the additional hashing layer H.

The following conclusions can be drawn from the results summarized in Table 3.

- **ADH/B:** Compared with ADH/B, ADH incurs small mAP decreases of 2.32%, 2.55%, 2.69% 3.99%, and 2.24% and a Rank-1 decreases of 4.45%, 6.51%, 4.61%, 3.45%, and 3.11% on the Market-1501, Market-1501+500k DukeMTMC-reID, CUHK03-NP (detected), and CUHK03-NP (labeled) datasets, respectively.

These findings are reasonable because binarizing high-dimensional real-value image descriptors into compact binary codes gives rise to a loss of information. The compact binary hashing codes allow storing a large number of codes in the RAM and efficiently computing the similarity with the Hamming distance, which enables a large-scale search. Hence, the trade-off of effectiveness and efficiency should be a consideration for practical applications.

- **ADH/Asy:** ADH/Asy results in impressive decreases of the mAP and Rank-1 in terms of absolute and relative degeneracy, which indicates that the asymmetric codes consistently boost the re-ID performance independently of the dataset. On Market-1501, the mAP drops 1.15% absolutely and 3.6% relatively and the Rank-1 also suffers from a drop of 3.33% absolutely and 5.7% relatively at 1024 bits without the asymmetry loss term, and ADH/Asy leads to very large mAP decreases of 1.15%, 1.68%, 1.09%, 2.37%, and 1.00% on the Market-1501, Market-1501+500k, DukeMTMC-reID, CUHK03-NP (detected), and CUHK03-NP (labeled) datasets, respectively.

The above results validate that when considering binary hashes to approximate similarity, much power can be gained by considering asymmetric codes even if the similarity measure is entirely symmetric, e.g., Hamming distance.

- **ADH/CE-v1:** Without the help of the point-wise cross-entropy loss, ADH/CE-v1 incurs huge mAP decreases of 3.23%, 3.93%, 3.61%, 1.2%, and 0.96% and Rank-1 decreases of 2.32%, 1.14%, 2.40%, 1.94%, and 2.14% on the Market-1501, Market-1501+500k, DukeMTMC-reID, CUHK03-NP (detected), and CUHK03-NP (labeled) datasets, respectively.

This result demonstrates that the classification loss is helpful for learning the hash mapping functions even just with simple linear classifiers to model the relationship between the learned binary codes and the label information, similar to our proposed method.

- **ADH/CE-v2:** For ADH/CE-v2, a slight performance deterioration compared with ADH also exists. Table 3 shows drops of 0.64%, 1.27%, 0.42%, 0.87%, and 0.28% in the mAP and 0.81%, 0.88%, 0.90%, 1.76%, and 1.85% in the Rank-1 on the Market-1501, Market-1501+500k, DukeMTMC-reID, CUHK03-NP (detected), and CUHK03-NP (labeled) datasets, respectively.

Although the two-stream framework can preserve the pairwise semantic information, the classification information in ADH/CE-v2 is only used to learn image representations, which is not fully exploited to have a direct impact on the hash mapping functions. By contrast, ADH is based on the assumption that the learned binary codes derived from the hash mapping functions should be optimal for the jointly learned linear classifiers, making full use of the label information.

Effect of the neural network architecture. To

Table 3 Performance analysis (R-1(%), mAP(%)) of different loss terms on the Market-1501, Market-1501+500k, DukeMTMC-reID, and CUHK03-NP datasets.

dim	Market-1501		Market-1501+500k		DukeMTMC-reID		CUHK03-NP(detected)		CUHK03-NP(labeled)	
	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP
ADH/B	63.12	34.25	53.54	23.16	68.37	34.06	36.67	34.12	38.23	34.21
ADH Ours	58.67	31.93	47.03	20.61	63.76	31.37	33.21	30.13	35.12	31.97
ADH/Asy	55.34	30.78	44.66	18.93	60.57	30.28	30.23	27.76	31.75	30.97
ADH/CE-v1	56.35	31.02	45.89	19.23	61.36	30.45	31.27	28.93	32.98	31.01
ADH/CE-v2	57.86	31.29	46.15	19.34	62.86	30.95	31.45	29.26	33.27	31.69

examine whether the proposed ADH works for other networks with different depths, widths, or topologies, and validate whether the generalization performance of it depends on the selection of the backbone, we conducted the following ablation experiments.

More recent architectures of the backbone network are used to repeat the same ablation study, which are divided into three categories in terms of model size. To be more specific, the compared small models are DenseNet[45], ShuffleNet-v1[46], ShuffleNet-v2[47], MobileNet-v2[48], and GhostNet[49]; the middle-sized models are the ResNeXt-50 + Elastic[50] and Big-littleNet[51]; and the large model is NASNet-A[52]. Note that we only experimented on the three largest datasets except CUHK03 because many backbones overfit on it due to its limited scale.

The re-ID performance of different backbones is summarized in Table 4, from which the following conclusions can be drawn.

- The re-Id performance varies from backbone to backbone, but the proposed ADH method is applicable to all compared backbones, indicating its robustness to deep neural networks with different architectures. Hence, a wide range of models can be adopted to increase the learning capacity.
- Superior re-ID performances can be obtained when using stronger backbones, which suggest that our ADH can readily benefit from further advancement of neural networks, a promising improvement of re-ID performance.
- Although the abundance of ablation studies provides a way to improve the re-ID performance in terms of the backbones, they also raise the proper tradeoff problem between the computational cost and re-ID accuracy. As shown in Table 4, the heavier backbones tend to have better performance but are also prone to increase the number of model parameters and floating point operations per second. Therefore, when the computation budget is fixed or when the model is easy to overfit as the training set is limited, it is of great importance to acquire acceptable accuracy under reasonable computational cost. Accordingly, the above table can be used as a reference.

4.3.4 Further analysis

In this section, we give more insights on the model design to motivate further study from the pointview of hashing code length, sensitivity to hyperparameters, impact of large search pools, and model testing efficiency.

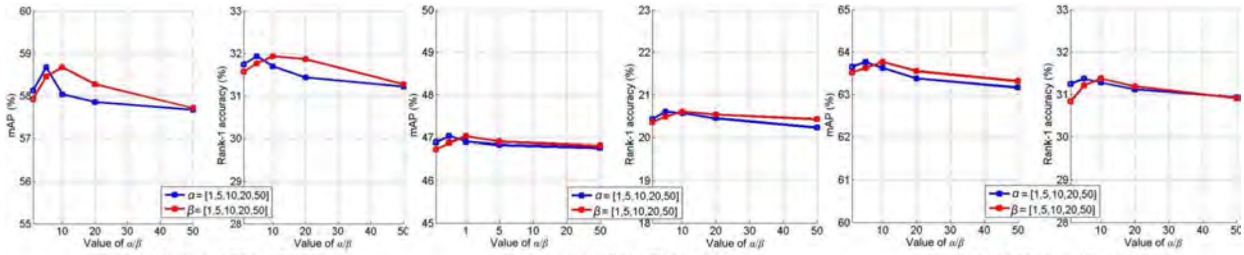
Sensitivity to hyperarameters. We evaluate the performance impact of the loss balance weights α and β in Eq. (8). Essentially, we tuned one parameter, with another parameter fixed. Specifically, we tuned α in the range of $[0, 10, 20, 30, 40, 50]$ by fixing $\beta = 10$. Similarly, we evaluated β in the same range with $\alpha = 5$.

The mAP scores and Rank-1 on the three compared datasets under the changes of different values of α and β are shown in Fig. 4. The hyperparameter α is not sensitive with a wide satisfactory range, similar to β. The proposed ADH can achieve satisfactory performance on the Market-1501, Market-1501+500k, and DukeMTMC-reID datasets $0 \leqslant \alpha \leqslant 50$. Thus, our model is insensitive to the hyperparameters and always achieves a satisfactory performance when α and β are in a wide range, which demonstrates the robustness and effectiveness of the proposed method.

Larger search pool. In this study, we conducedt experiments on a larger search pool to further demonstrate the superiority of the proposed method. Specifically, we enlarged the search pool with 34 574 person images from an auxiliary dataset[53], which acts as additional imposters and is independent of the Market-1501, Market-1501+500k, and DukeMTMC-

Table 4 Performance analysis of different backbones.

Backbone	Number of parameters ($\times10^6$)	Number of floating point operations per second ($\times10^6$)	Market-1501		Market-1501+500k		DukeMTMC-reID	
			R-1	mAP	R-1	mAP	R-1	mAP
DenseNet[45]	2.9	274	59.23	32.05	49.65	21.15	65.12	32.69
ShuffleNet-v1[46]	3.4	292	59.45	32.23	49.92	21.86	65.46	32.75
ShuffleNet-v2[47]	3.5	299	60.23	32.45	50.67	21.95	66.13	32.96
MobileNet-v2[48]	2.6	325	61.67	32.78	52.35	22.17	67.95	33.25
GhostNet[49]	5.2	141	63.42	35.11	52.99	24.43	68.97	35.17
ResNeXt-50 + Elastic[50]	25.2	4200	63.05	33.15	54.17	23.05	69.23	34.98
Big-littleNet[51]	26.2	2500	64.36	33.71	55.46	23.31	70.93	35.23
NASNet-A[52]	88.9	23 800	66.23	34.12	57.19	24.06	72.29	35.98
Ours	25.5	4090	58.67	31.93	47.03	20.61	63.76	31.37



(a) Market-1501-1024 bit (b) Market-1501+500k-1024 bit (c) DukeMTMC-reID-1024 bit

Fig. 4 mAP scores and Rank-1 with changes in parameters α and β when the hashing code length was set to 1024 on three datasets, i.e., Market-1501, Market-1501+500k, and DukeMTMC-reID.

reID datasets. Such a deployment is very consistent with real-world scenario.

We only evaluated the competitive hashing methods on this enlarged search pool given their fast search capability and low memory cost compared to conventional re-ID models. As shown in Table 5, all the compared methods suffer lower CMC and mAP performance due to the existence of imposters from the large search pool, but the proposed ADH remained its superior performance, which indicates the scalability and superiority of the proposed model in large-scale deployments.

Model testing efficiency. We made a comparison of model testing efficiency with three types of methods, i.e., type of handcrafted feature + metric learning, deep learning based re-ID, and hashing methods. Typical methods are selected: BoW + KISSME[29], IDE[54], PDH[14], SSDH[11], SDH[30], and X-ICE[16].

We implemented these experiments based on the open-source deep learning framework Pytorch as the platform with two NVIDIA TITAN XP GPUs, and the comparison results are summarized in Table 6. The testing time for

Table 5 Performance evaluation on large search pools with 34 574 imposters. The four partitionings are the unsupervised traditional hashing methods, supervised traditional hashing methods, deep learning based hashing methods, and hashing methods particularly designed for re-ID.

Method	Market-1501		Market-1501+500k		DukeMTMC-reID		CUHK03-NP(detected)		CUHK03-NP(labeled)	
	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP
SH[26]	8.13	4.21	7.05	5.34	9.83	5.25	3.25	4.11	3.81	8.11
ITQ[27]	8.23	4.35	7.26	5.43	10.11	5.31	3.37	4.23	3.95	8.23
DPLM[28]	19.11	11.23	17.25	9.54	21.23	11.34	10.11	10.01	10.12	11.02
SGH[29]	17.81	10.12	15.06	8.21	19.11	10.13	8.07	9.98	10.12	10.01
SDH[30]	16.36	9.23	14.56	7.21	17.67	9.11	10.15	9.01	9.81	9.72
DPSH[10]	40.12	20.11	36.54	19.15	32.45	16.51	26.25	25.11	26.11	25.92
ADSH[31]	48.67	25.12	42.16	17.05	49.17	26.12	28.23	28.11	29.16	28.75
DAPH[32]	42.23	23.02	38.65	22.72	33.05	16.51	26.07	24.93	26.61	25.13
Greedy Hash[33]	46.23	25.76	42.23	25.46	38.16	22.78	32.56	30.98	32.54	30.01
PDH[14]	44.13	24.93	35.21	15.11	48.95	26.11	28.04	27.91	29.93	28.87
X-ICE[16]	16.11	9.83	14.45	9.91	17.94	11.34	12.87	14.23	12.56	11.92
DRCSH[34]	21.85	11.11	19.42	12.11	23.56	13.57	14.51	13.98	15.09	18.82
Ours	**55.76**	**30.07**	**44.15**	**19.16**	**60.17**	**30.25**	**33.17**	**31.91**	**35.11**	**32.73**

Table 6 Comparison of the model testing time with three other typical methods on the Market-1501 dataset.

Method	Number of dimensions	Data type	Testing time (ms)			Total coding time (ms)
			Feature extraction	Distance calculation	Sorting	
BoW + kissme[29]	5600	Float	264.3	139.3	4.9	409.1
IDE[54]	4096	Float	8.3	97.9	3.5	109.7
PDH[14]	2048	Bool	32.8	0.98	0.83	34.61
SSDH[11]	1024	Bool	8.2	0.96	0.81	9.97
SDH[30]	1024	Bool	8.1	0.96	0.81	9.87
X-ICE[16]	1024	Bool	8.2	0.97	0.82	9.99
Ours	1024	Bool	**7.67**	**0.97**	**0.81**	**9.45**

each method was derived from three steps, i.e., feature extraction step, similarity calculation step, and sorting step. Moerover, the similarity of floating real-value data was calculated with the distance, whereas the similarity of bool data was calculated with the Hamming distance. The following conclusions can be drawn from Table 6.

- In terms of storage efficiency, the bool data have a great advantage over the floating data. Particulatly, the proposed ADH is superior as it can have a competitive re-ID performance but with much shorter hashing codes.
- The feature extraction time of the proposed ADH is 7.67 ms, which is acceptable for practical applications.
- The computation of the Hamming distance is much faster than that of the Euclidean distance, which further validates the advantage of the hashing method over real-value data in terms of efficiency.
- Generally, the sorting time of the Hamming distance based methods is much less than that of the Euclidean distance of the float-point feature-based one, which once again indicates that the hashing method is a choice for fast search in practice.
- In conclusion, from the viewpoint of efficiency, the hashing based methods are much better in terms of storage space and total searching time. Considering that re-ID tends to progress to large-scale evaluations, binary representations will become even more important. Particularly, the proposed ADH obtains considerable re-ID accuracy, which is largely attributed to the well-formulated asymmetry term of hashing codes. Moreover, the performance can be further improved with more semantic information such as part model[55, 56] or other more effective feature extractor[57]. Thus, we propose the asymmetric hashing method as a great balance of searching efficiency and acceptable accuracy.

5 Conclusion

In this study, we employed an effective deep hashing model for large-scale re-ID and proposed an ADH framework. The proposed ADH aims to generate pairwise similarity-preserving and semantic information-rich asymmetric hash functions. The two-stream deep neural networks simultaneously learn discriminative image representation and generate asymmetric hash mapping functions. Moreover, the proposed asymmetry regularizer between the float features and binary codes reduces the discrepancy between the real-value network output space and the desired Hamming space, boosting the final re-ID accuracy by a large margin. In addition, an efficient alternating optimization algorithm was devised for the formulated NP-hard problem. The experiments on four large-scale re-ID datasets show the effectiveness of the proposed ADH and its advantage over a wide range of state-of-the-art methods. Moreover, more complex and effective network structures can be easily employed as the backbone of our ADH, indicating promising re-ID performance boost. To the best of our knowledge, ADH is the first deep asymmetric hashing based method for large-scale re-ID. In future studies, we will investigate more extensions of asymmetric hashing codes, particularly in terms of efficiency and effectiveness.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. 61701277 and 61771288), the State Key Development Program in the 13th Five-Year Plan (No. 2017YFC0821601), and Open Project Fund of the National Engineering Laboratory for Intelligent Video Analysis and Application.

References

[1] B. Neyshabur, P. Yadollahpour, Y. Makarychev, R. Salakhutdinov, and N. Srebro, The power of asymmetry in binary hashing, arXiv preprint arXiv: 1311.7662, 2013.

[2] R. Zhao, W. L. Ouyang, and X. G. Wang, Person re-identification by salience matching, in *Proc. 2013 IEEE Int. Conf. Computer Vision*, Sydney, Australia, 2013, pp. 2528–2535.

[3] F. Xiong, M. R. Gou, O. Camps, and M. Sznaier, Person re-identification using kernel-based metric learning methods, in *Proc. 13th European Conf. Computer Vision*, Zurich, Switzerland, 2014, pp. 1–16.

[4] S. C. Liao, Y. Hu, X. Y. Zhu, and S. Z. Li, Person re-identification by Local Maximal Occurrence representation and metric learning, in *Proc. 2015 IEEE Conf. Computer Vision and Pattern Recognition*, Boston, MA, USA, 2015, pp. 2197–2206.

[5] M. Köstinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, Large scale metric learning from equivalence constraints, in *Proc. 2012 IEEE Conf. Computer Vision and Pattern Recognition*, Providence, RI, USA, 2012, pp. 2288–2295.

[6] M. Dikmen, E. Akbas, T. S. Huang, and N. Ahuja, Pedestrian recognition with a learned metric, in *Proc. 10th Asian Conf. Computer Vision*, Queenstown, New Zealand, 2010, pp. 501–512.

[7] R. R. Varior, B. Shuai, J. W. Lu, D. Xu, and G. Wang, A siamese long short-term memory architecture for human re-identification, in *Proc. 14th European Conf. Computer Vision*, Amsterdam, The Netherlands, 2016, pp. 135–153.

[8] L. M. Zhao, X. Li, Y. T. Zhuang, and J. D. Wang, Deeply-learned part-aligned representations for person re-identification, in *Proc. 2017 IEEE Int. Conf. Computer Vision*, Venice, Italy, 2017, pp. 3219–3228.

[9] Y. F. Sun, L. Zheng, W. J. Deng, and S. J. Wang, SVDNet for pedestrian retrieval, in *Proc. 2017 IEEE Int. Conf. Computer Vision*, Venice, Italy, 2017, pp. 3800–3808.

[10] W. J. Li, S. Wang, and W. C. Kang, Feature learning based deep supervised hashing with pairwise labels, arXiv preprint arXiv: 1511.03855, 2016.

[11] H. F. Yang, K. Lin, and C. S. Chen, Supervised learning of semantics-preserving hash via deep convolutional neural networks, arXiv preprint arXiv: 1507.00101, 2017.

[12] H. M. Liu, R. P. Wang, S. G. Shan, and X. L. Chen, Deep supervised hashing for fast image retrieval, in *Proc. 2016 IEEE Conf. Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016, pp. 2064–2072.

[13] Z. F. Qiu, Y. W. Pan, T. Yao, and T. Mei, Deep semantic hashing with generative adversarial networks, in *Proc. 40th Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, Shinjuku, Tokyo, Japan, 2017, pp. 225–234.

[14] F. Q. Zhu, X. W. Kong, L. Zheng, H. Y. Fu, and Q. Tian, Part-based deep hashing for large-scale person re-identification, *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4806–4817, 2017.

[15] L. Wu, Y. Wang, Z. Y. Ge, Q. C. Hu, and X. Li, Structured deep hashing with convolutional neural networks for fast person re-identification, *Comput. Vis. Image Underst.*, vol. 167, pp. 63–73, 2018.

[16] X. T. Zhu, B. T. Wu, D. C. Huang, and W. S. Zheng, Fast open-world person re-identification, *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2286–2300, 2018.

[17] K. Lin, H. F. Yang, J. H. Hsiao, and C. S. Chen, Deep learning of binary hash codes for fast image retrieval, in *Proc. 2015 IEEE Conf. Computer Vision and Pattern Recognition Workshops*, Boston, MA, USA, 2015, pp. 27–35.

[18] H. F. Yang, K. Lin, and C. S. Chen, Supervised learning of semantics-preserving hash via deep convolutional neural networks, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 437–451, 2018.

[19] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, Performance measures and a data set for multi-target, multi-camera tracking, in *Proc. European Conf. Computer Vision*, Amsterdam, The Netherlands, 2016, pp. 17–35.

[20] L. Zheng, L. Y. Shen, L. Tian, S. J. Wang, J. D. Wang, and Q. Tian, Scalable person re-identification: A benchmark, in *Proc. 2015 IEEE Int. Conf. Computer Vision*, Santiago, Chile, 2015, pp. 1116–1124.

[21] Z. Zhong, L. Zheng, D. L. Cao, and S. Z. Li, Re-ranking person re-identification with k-reciprocal encoding, in *Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 2017, pp. 1318–1327.

[22] W. Li, R. Zhao, T. Xiao, and X. Wang, Deepreid: Deep filter pairing neural network for person reidentification. in *Proceedings of Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 2014, pp. 152–159.

[23] D. Gray, S. Brennan, and H. Tao, Evaluating appearance models for recognition, reacquisition, and tracking, in *Proc. IEEE Int. Workshop on Performance Evaluation for Tracking and Surveillance*, Las Vegas, NV, USA, 2007, pp. 1–7.

[24] X. Fan, W. Jiang, H. Luo, and M. J. Fei, SphereReID: Deep hypersphere manifold embedding for person re-identification, *J. Vis. Commun. Image Represent.*, vol. 60, pp. 51–58, 2019.

[25] Y. Weiss, A. Torralba, and R. Fergus, Spectral hashing, in *Proc. 21st Int. Conf. Neural Information Processing Systems*, Vancouver, British Columbia, Canada, 2008, pp. 1753–1760.

[26] Y. C. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2916–2929, 2013.

[27] F. M. Shen, X. Zhou, Y. Yang, J. K. Song, H. T. Shen, and D. C. Tao, A fast optimization method for general binary code learning, *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5610–5621, 2016.

[28] Q. Y. Jiang and W. J. Li, Scalable graph hashing with feature transformation, in *Proc. 24th Int. Conf. Artificial Intelligence*, Buenos Aires, Argentina, 2018, pp. 2248–2254.

[29] F. M. Shen, C. H. Shen, W. Liu, and H. T. Shen, Supervised discrete hashing, in *Proc. 2015 IEEE Conf. Computer Vision and Pattern Recognition*, Boston, MA, USA, 2015, pp. 37–45.

[30] Q. Y. Jiang and W. J. Li, Asymmetric deep supervised hashing, in *Proc. 32nd AAAI Conf. Artificial Intelligence*, New Orleans, LA, USA, 2018.

[31] F. M. Shen, X. Gao, L. Liu, Y. Yang, and H. T. Shen, Deep asymmetric pairwise hashing, in *Proc. 25th ACM Int. Conf. Multimedia*, Mountain View, CA, USA, 2017, pp. 1522–1530.

[32] S. P. Su, C. Zhang, K. Han, and Y. H. Tian, Greedy hash: Towards fast optimization for accurate hash coding in CNN, in *Proc. 32nd Int. Conf. Neural Information Processing Systems*, Montréal, Canada, 2018, pp. 806–815.

[33] R. M. Zhang, L. Lin, R. Zhang, W. M. Zuo, and L. Zhang, Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification, *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 4766–4779, 2015.

[34] K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, Deep residual learning for image recognition, in *Proc. 2016 IEEE Conf. Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016, pp. 770–778.

[35] C. Jose and F. Fleuret, Scalable metric learning via weighted approximate rank component analysis, in *Proc. 14th European Conf. Computer Vision*, Amsterdam, The Netherlands, 2016, pp. 875–890.

[36] S. Karanam, M. R. Gou, Z. Y. Wu, A. Rates-Borras, O. Camps, and R. J. Radke, A comprehensive evaluation and benchmark for person re-identification: Features, metrics, and datasets, arXiv preprint arXiv: 1605.09653, 2018.

[37] Y. F. Sun, L. Zheng, Y. Yang, Q. Tian, and S. J. Wang, Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline), in *Proc. 15th European Conf. Computer Vision*, Munich, Germany, 2018, pp. 480–496.

[38] Y. B. Chen, X. T. Zhu, and S. G. Gong, Person re-identification by deep learning multi-scale representations, in *Proc. 2017 IEEE Int. Conf. Computer Vision*, Venice, Italy, 2017, pp. 2590–2600.

[39] S. C. Pang, S. B. Qiao, T. Song, J. L. Zhao, and P. Zheng, An improved convolutional network architecture based on residual modeling for person re-identification in edge computing, *IEEE Access*, vol. 7, pp. 106748–106759, 2019.

[40] H. Y. Wang, T. Fang, Y. L. Fan, and W. Wu, Person re-identification based on DropEasy method, *IEEE Access*, vol. 7, pp. 97021–97031, 2019.

[41] X. L. Qian, Y. W. Fu, T. Xiang, Y. G. Jiang, and X. Y. Xue, Leader-based multi-scale attention deep architecture for person re-identification, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 371–385, 2020.

[42] L. H. Wei, S. L. Zhang, H. T. Yao, W. Gao, and Q. Tian, GLAD: Global-local-alignment descriptor for scalable person re-identification, *IEEE Trans. Multimed.*, vol. 21, no. 4, pp. 986–999, 2019.

[43] K. Han, J. Y. Guo, C. Zhang, and M. J. Zhu, Attribute-aware attention model for fine-grained representation learning, in *Proc. 26th ACM Int. Conf. Multimedia*, Seoul, South Kerean, 2018, pp. 2040–2048.

[44] J. Y. Guo, Y. H. Yuan, L. Huang, C. Zhang, J. G. Yao, and K. Han, Beyond human parts: Dual part-aligned representations for person re-identification, in *Proc. Int. Conf. Computer Vision*, Seoul, Republic of Korea, 2019, pp. 3642–3651.

[45] G. Huang, S. C. Liu, L. van der Maaten, and K. Q. Weinberger, CondenseNet: An efficient densenet using learned group convolutions, in *Proc. 2018 IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 2752–2761.

[46] A. G. Howard, M. L. Zhu, B. Chen, D. Kalenichenko, W. J. Wang, T. Weyand, M. Andreetto, and H. Adam, MobileNets: Efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv: 1704.04861, 2017.

[47] N. N. Ma, X. Y. Zhang, H. T. Zheng, and J. Sun, ShuffleNet V2: Practical guidelines for efficient CNN architecture design, in *Proc. 15th European Conf. Computer Vision*, Munich, Germany, 2018, pp. 116–131.

[48] M. Sandler, A. Howard, M. L. Zhu, A. Zhmoginov, and L. C. Chen, MobileNetV2: Inverted residuals and linear bottlenecks, in *Proc. 2018 IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 4510–4520.

[49] K. Han, Y. H. Wang, Q. Tian, J. Y. Guo, C. J. Xu, and C. Xu, GhostNet: More features from cheap operations, in *Proc. 2020 IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2020, pp. 1580–1589.

[50] S. N. Xie, R. Girshick, P. Dollar, Z. W. Tu, and K. M. He, Aggregated residual transformations for deep neural networks, in *Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 2017, pp. 1492–1500.

[51] C. F. Chen, Q. F. Fan, N. Mallinar, T. Sercu, and R. Feris, Big-little net: An efficient multi-scale feature representation for visual and speech recognition, arXiv preprint arXiv: 1807.03848, 2019.

[52] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, Learning transferable architectures for scalable image recognition, in *Proc. 2018 IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 8697–8710.

[53] L. Zheng, Z. Bie, Y. F. Sun, J. D. Wang, C. Su, S. J. Wang, and Q. Tian, MARS: A video benchmark for large-scale person re-identification, in *Proc. 14th European Conf. on Computer Vision*, Amsterdam, The Netherlands, 2016, pp. 868–884.

[54] T. Xiao, S. Li, B. C. Wang, L. Lin, and X. G. Wang, Joint detection and identification feature learning for person search, in *Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 2017, pp. 3415–3424.

[55] Y. L. Zhao, Y. L. Li, and S. J. Wang, Person re-identification with effectively designed parts, *Tsinghua Science and Technology*, vol. 25, no. 3, pp. 415–424, 2020.

[56] Y. F. Sun, Z. P. Dou, Y. L. Li, and S. J. Wang, Improving semantic part features for person re-identification with supervised non-local similarity, *Tsinghua Science and Technology*, vol. 25, no. 5, pp. 636–646, 2020.

[57] Y. Li, X. M. Tao, and J. H. Lu, Effectively lossless subspace appearance model compression using prior information, *Tsinghua Science and Technology*, vol. 20, no. 4, pp. 409–416, 2015.

Yali Li received the BE degree with excellent graduates award from Nanjing University, China, in 2007 and the PhD degree from Tsinghua University, Beijing, China, in 2013. Currently she is a research assistant at the Department of Electronic Engineering, Tsinghua University. Her research interests include image processing, pattern recognition, computer vision, and video analysis.

Yali Zhao received the BE degree in electronic engineering from Sichuan University, China, in 2014. She is a PhD student at the Department of Electronic Engineering, Tsinghua University. Her research interests include image retrieval, classification, and person re-identification.

Shengjin Wang received the BE degree from Tsinghua University, China, and the PhD degree from Tokyo Institute of Technology, Tokyo, Japan, in 1985 and 1997, respectively. From 1997 to 2003, he was a member of the research staff with the Internet System Research Laboratories, NEC Corporation, Japan. Since 2003, he has been a professor with the Department of Electronic Engineering, Tsinghua University, where he is currently the director of the Research Institute of Image and Graphics. His current research interests include image processing, computer vision, video surveillance, and pattern recognition.