ADVERSARIAL BINARY CODING FOR EFFICIENT PERSON RE-IDENTIFICATION

Zheng Liu, Jie Qin, Annan Li∗, Yunhong Wang, and Luc Van Gool

1Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University Beijing, China 100191
2Inception Institute of Artificial Intelligence, UAE
3Computer Vision Laboratory, ETH Zurich, Switzerland
zhengliu@buaa.edu.cn, qinjiebuaa@gmail.com, liannan@buaa.edu.cn, yhwang@buaa.edu.cn, vangool@vision.ee.ethz.ch

ABSTRACT

Person re-identification (ReID) aims at associating persons with the same identity across different views/scenes. Most existing methods improve matching accuracy by proposing high-dimensional real-valued features to represent person images comprehensively. However, considering the increasing data scale in real-world applications, the storage and matching efficiencies should be paid attention to as well. In this paper, we propose a binary coding approach for efficient ReID, inspired by the recent advances in adversarial learning. Specifically, the proposed Adversarial Binary Coding (ABC) implicitly fits the feature distribution to the expected binary one by optimizing the Wasserstein distance. To further enhance the semantic discriminability of binary codes, we seamlessly embed the ABC into a similarity measuring deep neural network. By end-to-end learning the framework, compact and discriminative binary features are generated for efficient and accurate ReID. Extensive experiments on large-scale benchmarks demonstrate the superiority of our approach over the state-of-the-art methods in both efficiency and accuracy.

Index Terms— Person re-identification, binary coding, adversarial learning, deep learning

1. INTRODUCTION

Given one or multiple person images, person re-identification (ReID) aims to retrieve the person with the same identity from different viewpoints/scenes. ReID enables various potential applications, however, it still remains challenging due to the significant variations in poses, viewpoints and illuminations across different cameras. Numerous methods have been proposed to deal with the visual ambiguities by high-dimensional real-valued features [1, 2]. These methods have achieved noteworthy success in matching accuracy, however, they lead to extremely high costs of time and storage for large-scale data in the mean time.

Binary coding (i.e., hashing) has emerged for ReID to address the above efficiency issues. Previous binary coding based ReID methods can be divided into two categories: 1) The conventional methods (see Fig. 1(a)) learn discriminative Hamming spaces [3, 4] by matrix transformation based optimizations, which are unaffordable in computational costs due to the non-convex loss functions and high-dimensional features of large-scale data. 2) As shown in Fig. 1(b), the deep network based binary coding [5, 6, 7, 8] inserts hashing layers at the end of the networks. This straightforward scheme hardly eliminates the correlations between dimensions of outputs to satisfy the important principles of hashing (e.g. balancedness and independence [9]) for high-quality binary codes.

We argue that both types of the binary coding methods for ReID lack a data-driven mechanism. These methods adopt external hashing constraints, such as $\ell_1$ and $\ell_2$ losses on each
dimension, to explicitly restrict the form of features, which is significantly harmful to the discriminability of features. They focus on the approximate likelihood of the limited examples and need to fiddle with additional noise terms for generalization. To overcome the above drawbacks, we propose to jointly learn a discriminative feature representation, an accurate similarity measurement and an implicit binary transformation. Instead of adopting explicit hashing constraints on each sample, we propose Adversarial Binary Coding (ABC) to directly measure the distance between the feature distribution and a prior binary data distribution inspired by the adversarial learning [10, 11, 12]. Furthermore, we embed the ABC into a network optimized by a triplet loss\(^1\). The deep network learns to generate binary features driven by the real binary data, while preserving semantic information for similarity measuring by the triplet loss in an unified low-dimensional space. Thus we correlates the compactness and discriminability of binary features in our framework for efficient and accurate ReID.

Our main contributions are summarized as follows: 1) We propose a binary transformation strategy (i.e. ABC) that fits the feature distribution to the binary one based on deep adversarial learning. 2) An end-to-end deep neural network is built for efficient ReID by seamlessly accommodating the ABC module for preserving semantics while binarizing features. 3) We present an adaptive mechanism for weighting the two losses to lay particular emphasis on either of the two losses at different training phases. 4) Extensive experiments on three large-scale ReID benchmarks clearly show the superiority of our framework.

2. APPROACH

2.1. Adversarial Binary Coding

As illustrated in Fig. 2, we develop our binary coding approach based on an adversarial learning framework, where a generator and a discriminator compete with each other. In particular, the feature extractor acts as the generator by extracting binary-like feature vectors to confuse the discriminator. Meanwhile, a sampler randomly and independently samples 0 and 1 values with 50% chance from the Bernoulli distribution to form binary codes, conforming to the hashing principles proposed in [9]. The discriminator is expected to distinguish whether a sample is from the real-valued feature distribution or the binary one. In other words, the discriminator aims to classify the binary codes as positive samples and the real-valued feature vectors as negative samples.

Formally, we denote a batch of \( n \) images as \( \mathbf{I} = \{ \mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_n \} \) under the distribution \( p_I \). The feature extractor (i.e. the generator) is denoted as a mapping function \( f(\mathbf{I}) \) under an encoding distribution \( q(Z \mid \mathbf{I}) \), where

\[^{1}\text{We adopt a triplet network in this study as a baseline. Note that the ABC can be flexibly embedded into any similarity measuring network.}\]

\[ Z = \{ z_1, z_2, \ldots, z_n \} \] denotes the extracted feature vectors. \( q(Z \mid \mathbf{I}) \) aims to transform data from the data distribution \( p_I \) to the feature distribution \( q \):

\[
q(z_i) = \int_{I_i} q(z_i \mid I_i) p_I(I_i) dI_i, \ i = 1, \ldots, n. \tag{1}
\]

As multiple Bernoulli samplings with the same probability is equivalent to a binomial distribution, the extractor is essentially regularized by fitting the posterior \( q \) to a prior binomial distribution \( b \) using the Wasserstein distance formulated as:

\[
\max_b \mathbb{E}_{B \sim p_B}[D_B(B, \hat{B})] - \mathbb{E}_{B \sim q_b}[D_B(B)] \tag{2}
\]

where \( D_B \) denotes the discriminator network with parameters \( \theta \), \( B \in \{0, 1\}^m \) denotes a randomly sampled binary code from \( b \), and \( m \) denotes the bit length. The Wasserstein distance has the unique advantage over other distance measurements (e.g. Jensen-Shannon (JS) and Kullback-Leibler (KL) divergence) that it measures the distances even if two distributions are totally non-overlapped, which has been theoretically analyzed in [13]. In our case, \( b \) corresponds to a set of discrete points (with 0 or 1 values), and \( q \) can be a random manifold in the original feature space. Before training, the distributions \( b \) and \( q \) can hardly have any intersection in the high-dimensional space. Therefore, by adopting the Wasserstein distance, it is theoretically guaranteed that the real-valued feature distribution could be fitted to the target binary distribution after adversarial training.

In terms of the discriminator, traditional methods usually adopt the fully-connected (FC) networks. However, the FC operation directly observes the entire vector at once and ignores the local pattern of binary codes (i.e. every dimension is a binary bit). This makes later layers hard to learn the different properties between binary and real-valued vectors. Moreover, the FC network results in a large number of parameters, especially for longer bit lengths in our task, which is harmful to the balance between the extractor and discriminator during training. Therefore, we adopt 1-D Convolution (Conv1D) to address the above issues. Conv1D observes several local dimensions of input vectors at once and gradually expands the receptive field to the entire vectors. This bottom-up strategy enables the discriminator to recognize the local properties of inputs. In addition, Conv1D requires much less parameters.
than FC layers, and thus the discriminator is much easier to optimize.

Current deep networks (e.g. ResNet [14]) commonly adopt Rectified Linear Unit (ReLU) as the activation function. Hence, we represent every bit of the binary codes by \{0, 1\} instead of \{-1, 1\} [6, 4] due to the non-negative outputs of ReLU. Generally, parameters in deep networks are initialized to values much smaller than 1, and are carefully controlled (e.g. by learning rates and weight decay mechanism) to avoid gradient vanishing or exploding. As a result, the features will also be very close to 0 since they share the same scale with weights. On the contrary, the ABC expects every dimension to be constrained near 0 or 1. Therefore, the optimization process will be unstable without normalization due to this contradiction.

To address the above issue, we normalize both the features and binary codes to the same scale by \(\ell_2\) normalization. As for the real-valued features, we adopt the standard \(\ell_2\) normalization. The \(\ell_2\) normalization for a batch of random binary vectors \(\{B_i\}_{i=1}^n\) is as follows:

\[
\bar{B}_i = \frac{B_i}{\|B_i\|_2}, \quad i = 1, \ldots, n. \tag{3}
\]

However, the \(\ell_2\)-Norm of every vector could be different since they may contain different numbers of non-zero entries assigned by random sampling. This leads to a non-deterministic target distribution, and further results in an unstable training process. Therefore, in this study, we adopt the expectation of the Bernoulli random variables \(\mathbb{E}[\text{Bernoulli}[1]]\) to calculate the \(\ell_2\)-Norm factor \(\lambda\) as:

\[
\lambda = \mathbb{E}_{\text{Bernoulli}[1]}[I_1^2]^{\frac{1}{2}}. \tag{4}
\]

Thus the binary vectors can be normalized as:

\[
\bar{B}_i = \frac{1}{\lambda} B_i, \quad i = 1, \ldots, n. \tag{5}
\]

### 2.2. Triplet Loss Based Efficient ReID Framework

To not only transform features to binary form, but also measure similarities between binary codes, the ABC is further embedded into a triplet network for ensuring the discriminability of the learned binary features. The triplet loss [15] \(L\) is formulated as follows:

\[
L(x_i, x_j, y_k, \alpha) = \frac{1}{n} \max\{d(x_i, x_j) - d(x_i, y_k) + \alpha, 0\}, \tag{6}
\]

where \(x_i\) and \(x_j\) are the features from the same identity, \(y_k\) is the feature from another identity, \(\alpha\) is the imposed distance margin between positive and negative pairs, and \(d(\cdot)\) measures the similarity distance.

The overall framework is shown in Fig. 3. ResNet-50 is adopted in our framework as the backbone model, in which the fixed average pooling layer is replaced by an adaptive average pooling to fit different input sizes, and a feature embedding (fully-connected) layer follows to reduce feature dimensions into expected lengths. At the beginning of training, we fine-tune the model on pedestrian images by solving an ID classification problem with Cross Entropy Error (CEE) loss to find a relative optimal initialization and boost the convergence. Subsequently, we train the model with normalization by jointly optimizing the Wasserstein loss for binary coding and triplet loss for similarity measuring.

During testing, the features are binarized as follows:

\[
b_j = \begin{cases} 0 & z_j \leq \frac{1}{\lambda}, j = 1, \ldots, m, \\ 1 & z_j > \frac{1}{\lambda}, \end{cases} \tag{7}
\]

where \(z_j\) is the value of the \(j\)-th entry of a real-valued feature \(z_i \in \mathbb{R}^m\) extracted by \(f(I_1)\), and \(b_j \in \{0, 1\}\) is the binary bit of \(z_j\) after binarization. The Hamming distances between queries and the gallery set are further computed using extremely fast XOR operations to measure similarities.

**Global Hard Negative Mining in Triplets:** Hard negative mining (HNM) methods [16] have been proposed to boost and improve the training of the triplet loss. Rather than the conventional approaches which usually exploit hard samples from local triplet batches, we introduce a global HNM strategy in this study. In each training epoch, every probe image is selected to be the anchor in a triplet. We first randomly pick an image of same identity from another view as the positive sample, and an image of different identity as the negative sample for several successive epochs. We call these epochs the **review period**, in which the model reviews samples as many as possible and records the nearest negative sample and the farthest positive sample of every anchor as the hardest samples. During the subsequent few epochs, which is called the **reinforce period**, the positive and negative samples for every anchor are fixed to improve the model on the hard triplets. Thus our HNM scheme can improve the performance by alternatively and globally updating the hardest samples and re-inforcing the model on tough triplets.

![Fig. 3. Illustration of network structure and the adversarial binary coding embedded efficient ReID framework.](image-url)
2.3. Adaptive Weight Adjustment

During training, we aim to maintain the relative balance between the adversarial loss and the triplet loss. If the triplet loss is significantly weaker, the binary features may lose the discriminability. Otherwise, the binarization may be failed. Hence, adaptively adjusting the weight of losses is helpful for the sustainable training progress.

The Wasserstein loss is formulated in [13] as a supremum related to a parameterized family of discriminator. Thus, the loss only provide the relative trend of the optimization instead of absolute progress in practice. To address this issue, we first formulate an indicator to evaluate the progress of binarization. We let \( z_t = [z_1, \ldots, z_m] \in \mathbb{R}^m \) indicate a batch of \( n \) feature vectors. The purpose of binarization is to push every bit away from the mean value \( \frac{1}{\sqrt{n}} \) after \( \ell_2 \) normalization. Therefore, we adopt the distance between the real-valued features and the mean vector to evaluate the degree of binarization as:

\[
E_1 = \frac{1}{\sqrt{m}} \beta \sum_{i=1}^{n} ||z_i - \beta||_2,
\]

where \( \beta \in \mathbb{R}^m \) is the mean vector in which every bit is \( \frac{1}{\sqrt{n}} \). \( E_1 \) is larger if the feature values are closer to binary bits, and reaches the supremum 0.5 when the features are absolute binary codes. However, using Eq. (8) alone cannot capture the correlation between dimensions, e.g. all the values in a vector may be close to 0. Therefore, we further formulate an additional term to evaluate the dispersion of values as:

\[
E_2 = \frac{\lambda}{n} \sum_{i=1}^{n} \frac{1}{m} \sum_{j=1}^{m} (z_{ij} - \frac{1}{2\sqrt{n}})^2.
\]

If the values in \( z_t \) disperse as expected, \( E_2 \) will be small. \( E_2 \) reaches the supremum 0.5 when the features are \( \mathbf{0} \) or \( \mathbf{1} \) vectors before normalization. Hence, we formulate the score of binarization as:

\[
E_W = \frac{1}{2(e^{\beta} - 1)}(e^{E_1} + e^{-E_2} - 2).
\]

Thus \( E_W \) locates in the range of \([0, 1]\). To evaluate the performance of similarity measuring, we let \( d_p^t \) denote the hardest positive distance in a triplet \( t \), and let \( d_n^t \) denote the hardest negative distance after the latest review period. The score of similarity measuring is formulated as:

\[
E_T = \frac{1}{N_T} \sum_{t=1}^{N_T} e^{-\min(d_p^t, d_n^t)/\alpha},
\]

where \( N_T \) is the number of triplets and \( \alpha \) is the margin. Thus the larger \( E_T \) indicates that the gap between positive and negative distances is closer to the margin, and \( E_T \) equals 1 when the gap is not less than the margin. Then we formulate the weights of the two losses as follows:

\[
w_W = \gamma \cdot \frac{E_W}{E_T + E_W},
\]

\[
w_T = 1 - w_W,
\]

where \( \gamma \) is a scale factor, and \( w_W \) and \( w_T \) indicate the weights for the Wasserstein and triplet losses, respectively.

3. EXPERIMENTS

3.1. Datasets and Settings

CUHK03 [17] contains 14,096 images of 1,467 identities under six cameras. The dataset provides both manually labeled (CUHK03-L) and detected (CUHK03-D) bounding boxes. We resize the images to \( 160 \times 60 \), and adopt the protocol reported by [17]. The margin of triplet loss on this dataset is initialized to 0.2 and increased to 0.3 after 1,000 epochs, 0.4 after 2,500 epochs, and 0.5 after 4,000 epochs.

Market-1501 [18] contains 32,668 detected \( 128 \times 64 \) images of 1,501 pedestrians under six cameras and provides a fixed evaluation protocol. The margin of triplet loss on this dataset is initialized to 0.2 and increased to 0.3 after 1,000 epochs and 0.4 after 4,000 epochs.

DukeMTMC-reID [19, 20] contains 36,411 images of 1,404 identities under 8 cameras and provides a fixed protocol. We resized images to \( 128 \times 64 \), and the margin of triplet loss on this dataset is initialized to 0.2 and increased to 0.3 after 2,000 epochs and 0.4 after 5,000 epochs.

We implement our framework using the PyTorch library on a PC with Intel Core CPUs (3.4GHz), 32 GB memory, and one NVIDIA GTX TITAN X GPU. We horizontally flip images for data augmentation and set the batch size to 64. The learning rate of the ResNet-50 is initialized as 0.001 and decreased to 0.0001 with the iterations. The learning rate of the discriminator is consistently set to 0.01. The review period and reinforce period are 20 epochs and 5 epochs, respectively. As for the adaptive weight adjustment, we set \( \gamma = 0.7 \). We re-run the comparison methods if the codes are publicly available to fairly evaluate their efficiencies.

3.2. Evaluation of Computation and Storage Efficiency

We first evaluate the time of querying (Q. Time) and the memory storing all the gallery features (Mem.) with different bit lengths. As can be seen in Fig. 4, the query time and memory consumed by binary codes are far less than those of real-valued features. Compared with the real-valued feature which generally requires 32-bit for storing the value of a dimension, the binary feature only requires 1-bit for every dimension. During testing, the bit-wise XOR operation is significantly faster than the real-valued Euclidean distances computation. As shown in Fig. 4, the longer the bit lengths,
Table 1. Matching rates (%) and mAP (%) by comparing with hashing methods.

| Method (512-bit) | CUHK03-L R-1 mAP | Market-1501 R-1 mAP | DukeMTMC-reID R-1 mAP | R-5 mAP | R-20 mAP |
|-----------------|------------------|---------------------|----------------------|---------|---------|
| DRSCH [6]       | 55.5             | 42.9                | 20.3                 | 47.2    | 33.1    |
| CSBT [4]        | 44.6             | 24.3                | -                    | -       | -       |
| PDH [7]         | -                | -                   | -                    | -       | -       |
| DeepSSH [8]     | -                | -                   | 24.1                 | -       | -       |
| DSRH* [5]       | 16.2             | 27.1                | 17.7                 | 25.6    | 18.6    |
| HashNet* [21]   | 29.4             | 29.2                | 19.1                 | 40.8    | 28.6    |
| DCH* [22]       | 44.4             | 41.3                | 30.7                 | 57.4    | 37.3    |
| ABC+Tri.+AWA    | 59.2             | 54.3                | 49.4                 | 66.5    | 59.9    |

Table 2. Comparison with state-of-the-art non-hashing methods on CUHK03-L.

| Method          | R-1 mAP | R-5 mAP | R-20 mAP | Q. Time | Mem. |
|-----------------|---------|---------|----------|---------|------|
| EDM [23]        | 51.9    | 53.6    | -        | -       | -    |
| SIR+CIR [24]    | 52.2    | -       | -        | -       | -    |
| SSM [2]         | 72.7    | 92.4    | -        | -       | -    |
| BraidNet [25]   | 85.8    | 98.5    | 99.6     | -       | -    |
| HAP2S [26]      | 88.9    | 98.4    | -        | -       | -    |
| ABC+Tri.+AWA    | 72.4    | 93.3    | 98.9     | 68.5    | 69.9 |

Table 3. Comparison with state-of-the-art non-hashing methods on Market-1501.

| Method          | R-1 mAP | R-5 mAP | Q. Time | Mem. |
|-----------------|---------|---------|---------|------|
| SIR+CIR [24]    | 57.5    | 59.5    | 1.08e+02| -    |
| SSM [2]         | 88.1    | 68.7    | 6.64e-01| -    |
| BraidNet [25]   | 83.7    | 69.5    | -       | -    |
| HAP2S [26]      | 84.6    | 69.4    | -       | -    |
| ABC+Tri.+AWA    | 81.3    | 61.2    | 4.3e+00 | -    |

Table 4. Comparison with state-of-the-art non-hashing methods on DukeMTMC-reID.

| Method          | R-1 mAP | R-5 mAP | Q. Time | Mem. |
|-----------------|---------|---------|---------|------|
| LOMO+XQDA [28]  | 50.8    | 51.5    | 1.08e+03| -    |
| CamStyle [27]   | 75.3    | 53.5    | 4.76e-01| -    |
| BraidNet [25]   | 76.4    | 59.5    | -       | -    |
| HAP2S [26]      | 75.9    | 60.6    | -       | -    |
| ABC+Tri.+AWA    | 82.5    | 61.2    | 4.3e+00 | -    |

the more resources and time we can save.

3.3. Comparison with Hashing Based Methods

We compare our framework with the following state-of-the-art binary coding (hashing) based ReID methods reported on the three datasets, including DRSCH [6], CSBT [4], PDH [7] and DeepSSH [8]. The results are shown in Table 1, where the best performance is highlighted in bold. We can observe that our method significantly outperforms the state-of-the-art hashing based ReID methods.

We also compare our method with several recently proposed hashing methods (noted by * in Table 1) that are not originally applied for ReID, including DSRH [5], HashNet [21] and DCH [22]. In DSRH, binary codes are directly obtained by a sign(·) function. HashNet adopts the Tanh activation for binarization, while DCH adopts an improved Sigmoid activation. For fairness, we employ the original network structure of these methods under the triplet loss with the 512 bit length. As can be seen, our method achieves much better results than the existing hashing methods.

3.4. Comparison with Non-Hashing ReID Methods

We compare our method under the best setting (2048-bit ABC+Tri.+AWA) with several non-hashing ReID methods. Since there is no attention mechanism involved in our framework, we mainly compare our method with the state-of-the-art non-hashing methods which do not explicitly use attention models or part-based schemes. The methods mainly include:

1) Deep Learning based methods, such as SIR+CIR [24], EDM [23], CamStyle [27], HAP2S [26], and BraidNet [25];

2) Non-deep methods, such as XQDA [28], and SSM [2].

The comparison results are shown in Tables 2, 3, and 4, respectively. It is obvious that our framework not only achieves competitive accuracies, but also achieves significant advantages in terms of the matching efficiencies. The advantages of performance in both accuracy and efficiency are more outstanding on large-scale Market-1501 and DukeMTMC-reID datasets. The results demonstrate that our method significantly close the gap of matching accuracies between hashing and non-hashing ReID methods.

3.5. Ablation Study

In this part, we analyze the effectiveness and necessity of our adversarial learning mechanism in Table 5. We first evaluate several baselines which remove the ABC module and minimize the ℓ1/2 distances between feature dimensions and binary values. As shown, the baselines fail to generate discriminative binary features due to the weak generalization capability. By contrast, our adversarial learning based binary coding cares more about fitting the distributions in a global feature space for high-quality binary codes.

In addition, we evaluate the effectiveness of adopting ℓ2 normalization (Norm.) of features and adaptive weight adjustment. The results in Table 5 demonstrate that the feature normalization successfully improves our method by addressing the contradiction between gradient scales of different losses. We can also observe from Table 5 that the adaptive weight adjustment successfully coordinates two losses and further improves the performance. Moreover, by comparing the results of using siamese [23] (Siam.) and triplet networks, we can find that our ABC module can boost the performance irrespective of network architectures.

4. CONCLUSION

In this work, we proposed a binary coding approach for efficient ReID by adopting an adversarial learning scheme. The proposed approach fitted the real-valued feature distribution to the binary one implicitly by optimizing the Wasserstein distance. To preserve the semantic discriminability, we equipped...
the adversarial binary coding network with a similarity measuring triplet network. The binary coding and similarity measuring networks were jointly optimized in an end-to-end manner. An adaptive mechanism was further incorporated for the equilibrium of training. Extensive experiments show that our method significantly outperformed the state-of-the-art hashing based approaches, and was competitive to the state-of-the-art ReID approaches in accuracies, whilst reducing computation and memory costs significantly.

5. REFERENCES

[1] Jiaxin Chen, Zhaoxiang Zhang, and Yunhong Wang, “Relevance metric learning for person re-identification by exploiting listwise similarities,” IEEE TIP, vol. 24, no. 12, pp. 4741–4755, 2015.

[2] Song Bai, Xiang Bai, and Qi Tian, “Scalable person re-identification on supervised smoothed manifold,” in CVPR, 2017.

[3] Feng Zheng and Ling Shao, “Learning cross-view binary identities for fast person re-identification,” in IJCAI, 2016.

[4] Jiaxin Chen, Yunhong Wang, Jie Qin, Li Liu, and Ling Shao, “Fast person re-identification via cross-camera semantic binary transformation,” in CVPR, 2017.

[5] Fang Zhao, Yongzhen Huang, Liang Wang, and Tieniu Tan, “Deep semantic ranking based hashing for multi-label image retrieval,” in CVPR, 2015.

[6] Ruimao Zhang, Liang Lin, Rui Zhang, Wangmeng Zuo, and Lei Zhang, “Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification,” IEEE TIP, vol. 24, no. 12, pp. 4766–4779, 2015.

[7] Fuqing Zhu, Xiangwei Kong, Liang Zhang, Haiyan Fu, and Qi Tian, “Part-based deep hashing for large-scale person re-identification,” IEEE TIP, vol. 26, no. 10, pp. 4806–4817, 2017.

[8] Ya Zhao, Sihui Luo, Yezhou Yang, and Mingli Song, “Deepssh: Deep semantic structured hashing for explainable person re-identification,” in IJCV, 2018, pp. 1653–1657.

[9] Yair Weiss, Antonio Torralba, and Rob Fergus, “Spectral hashing,” in NIPS, 2009.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, “Generative adversarial nets,” in NIPS, 2014.

[11] Alec Radford, Luke Metz, and Soumith Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” arXiv preprint arXiv:1511.06434, 2015.

[12] Song Bai, Yingwei Li, Yuyin Zhou, Qizhu Li, and Philip HS Torr, “Adversarial metric attack for person re-identification,” arXiv preprint arXiv:1901.10650, 2019.

[13] Martin Arjovsky, Soumith Chintala, and Léon Bottou, “Wasserstein gan,” arXiv preprint arXiv:1701.07875, 2017.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” in CVPR, 2016.

[15] Vassilios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk, “Learning local feature descriptors with triplets and shallow convolutional neural networks,” in BMVC, 2016.

[16] Ergys Ristani and Carlo Tomasi, “Features for multi-target multi-camera tracking and re-identification,” in CVPR, 2018.

[17] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang, “Deepreid: Deep filter pairing neural network for person re-identification,” in CVPR, 2014.

[18] Liang Zheng, Liyue Shen, Lu Tian, Shengjun Wang, Jingdong Wang, and Qi Tian, “Scalable person re-identification: A benchmark,” in ICCV, 2015.

[19] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi, “Performance measures and a data set for multi-target, multi-camera tracking,” in ECCVW, 2016.

[20] Zhedong Zheng, Liang Zheng, and Yi Yang, “Unlabeled samples generated by gan improve the person re-identification baseline in vitro,” in ICCV, 2017.

[21] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Philip S Yu, “Hashnet: Deep learning to hash by continuation,” ICCV, 2017.

[22] Yue Cao, Mingsheng Long, Liu Bin, and Jianmin Wang, “Deep cauchy hashing for hamming space retrieval,” in CVPR, 2018.

[23] Hailin Shi, Yang Yang, Xiangyu Zhu, Shengcai Liao, Zhen Lei, Weishi Zheng, and Stan Z Li, “Embedding deep metric for person re-identification: A study against large variations,” in ECCV, 2016.

[24] Faqiang Wang, Wangmeng Zuo, Lin Li, David Zhang, and Lei Zhang, “Joint learning of single-image and cross-image representations for person re-identification,” in CVPR, 2016.

[25] Yicheng Wang, Zhenzhong Chen, Feng Wu, and Gang Wang, “Person re-identification with cascaded pairwise convolutions,” in CVPR, June 2018.

[26] Rui Yu, Zhiyong Dou, Song Bai, Zhaohui Yang, Yongchao Xu, and Xiang Bai, “Hard-aware point-to-set deep metric for person re-identification,” in ECCV, September 2018.

[27] Zhen Zhong, Liang Zheng, Zhedong Zheng, Shaozi Li, and Yi Yang, “Camera style adaptation for person re-identification,” in CVPR, June 2018.

[28] Shengcai Liao, Yang Hu, Xiangyu Zhu, and Stan Z Li, “Person re-identification by local maximal occurrence representation and metric learning,” in CVPR, 2015.

Table 5. Comparison with different baselines of our framework. The bit length is 2,048.

| Method            | CUHK01-L rank1 mAP | CUHK01-L rank1 mAP | DukeMTMC-reID rank1 mAP | DukeMTMC-reID rank1 mAP |
|-------------------|---------------------|---------------------|-------------------------|-------------------------|
| Real-valued Triplet Siamese | 76.9 74.4 71.3 68.9 | 79.5 58.7 52.5 43.8 | 81.4 61.7 55.0 44.3 |
| Tri+Sigmoid+db | 82.3 79.6 75.2 71.0 | 84.2 71.0 63.0 44.3 | 86.5 72.1 65.2 43.8 |
| Tri+ABC | 80.5 77.8 73.6 69.4 | 82.2 70.2 60.9 44.1 | 84.0 69.1 61.2 44.3 |
| Tri+ABC+AWA | 81.0 78.3 74.2 70.0 | 83.0 71.0 62.0 44.2 | 85.0 70.5 62.5 44.3 |
| Tri+ABC+A-Sigmoid | 76.0 77.2 72.8 69.2 | 78.0 70.0 63.2 44.3 | 80.0 68.0 60.5 44.3 |

705