Learning Transformation-Invariant Local Descriptors With Low-Coupling Binary Codes

Miao, Yunqi; Lin, Zijia; Ma, Xiao; Ding, Guiguang; Han, Jungong

Published in:
IEEE Transactions on Image Processing

DOI:
10.1109/TIP.2021.3106805

Publication date:
2021

Citation for published version (APA):
Miao, Y., Lin, Z., Ma, X., Ding, G., & Han, J. (2021). Learning Transformation-Invariant Local Descriptors With Low-Coupling Binary Codes. *IEEE Transactions on Image Processing*, *30*, 7554 - 7566. https://doi.org/10.1109/TIP.2021.3106805

Learning Transformation-Invariant Local Descriptors with Low-Coupling Binary Codes

Yunqi Miao, Zijia Lin, Xiao Ma, Guiguang Ding, *Member, IEEE* and Jungong Han

***Abstract*—Despite the great success achieved by prevailing binary local descriptors, they are still suffering from two problems: 1) vulnerable to the geometric transformations; 2) lack of an effective treatment to the highly-correlated bits that are generated by directly applying the scheme of image hashing. To tackle both limitations, we propose an unsupervised Transformation-invariant Binary Local Descriptor learning method (TBLD). Specifically, the transformation invariance of binary local descriptors is ensured by projecting the original patches and their transformed counterparts into an *identical* high-dimensional feature space and an *identical* low-dimensional descriptor space simultaneously. Meanwhile, it enforces the dissimilar image patches to have distinctive binary local descriptors. Moreover, to reduce high correlations between bits, we propose a bottom-up learning strategy, termed *Adversarial Constraint Module*, where low-coupling binary codes are introduced externally to guide the learning of binary local descriptors. With the aid of the Wasserstein loss, the framework is optimized to encourage the distribution of the generated binary local descriptors to mimic that of the introduced low-coupling binary codes, eventually making the former more low-coupling. Experimental results on three benchmark datasets well demonstrate the superiority of the proposed method over the state-of-the-art methods. The project page is available at https://github.com/yoqim/TBLD.**

***Index Terms*—binary local descriptor, patch matching, deep learning.**

I. Introduction

A Local descriptor is used to characterize the region around an interest point in an image, *i.e.,* image patch. Local descriptors are widely applied in visual tasks like visual search [1], object recognition [2] and face recognition [3], [4], *etc*. Therefore, learning effective local descriptors has become an active topic in the community of computer vision. Recently, binary local descriptors, due to the high compactness and high matching speed, have become prevalent for applications with large-scale data.

Over the past decade, numerous binary local descriptors have been proposed, including hand-crafted ones (BRISK [5], BRIEF [6], ORB [7], *etc*.), and learning-based ones (Binboost [8], LDAHash [9], *etc*.). Inspired by the advances of deep learning techniques, deep learning approaches for binary local descriptors have recently drawn increasing attention, like DeepBit [10], DBD-MQ [11], L2-Net [12], and GraphBit [13]. Depending on whether the labeled data are required, deep binary local descriptors can be further categorized as supervised [12], [14], [15] and unsupervised [10], [11], [13], [16] ones. Supervised methods generally achieve better performance with the supervision given by pairwise labels, indicating whether two patches come from the same category or not. However, such pairwise labels are too expensive to obtain in real-world applications. Therefore, unsupervised learning methods have gained more attention recently. Despite the remarkable performance improvements, there are still problems that need to be better addressed.

Firstly, an effective binary local descriptor should be robust against geometric transformations, *i.e.,* rotation, scaling, and viewpoint changes. The robustness of local descriptors will affect the matching accuracy in matching/retrieval tasks [17]. Earlier binary local descriptors [5]–[7] are built upon hand-crafted sampling patterns or pairwise intensity comparisons, which are vulnerable to geometric distortions due to the high sensitivity of hand-crafted features. Thus, hand-crafted binary local descriptors tend to have unstable performances [10]. On the other hand, most existing deep unsupervised binary local descriptors focus more on generating effective compact codes but pay little attention to the robustness against geometric transformations [11], [13]. A prior work, DeepBit [10], enhances the robustness of the descriptor against rotation via minimizing the Hamming distance between the descriptors of an original image and its transformed counterparts. Although it provides an intuitive way to generate the transformation-invariant local descriptors, a problem might be that such work is based on the idea that an original image and its transformed counterparts should be represented by different descriptors. However, ideally, the same object is expected to be described by exactly the same descriptor, regardless of the viewpoint or distance changes. Therefore, simply minimizing the distance between the original image and its transformed counterparts is not the optimal solution.

Secondly, an effective binary local descriptor is supposed to be informative, *i.e.,* each bit carrying distinctive information. However, previous learning-based descriptors generally follow the scheme of image hashing. Yet an image patch, as a small region around an interest point, generally contains much less information than an image. Therefore, directly employing image hashing schemes can probably lead to highly-correlated bits, which means information contained in different bits

Manuscript received February xxx; revised xxx and xxx; accepted xxx, 2018. This work was partially supported by the National Natural Science Foundation of China (Nos. U1936202, 61925107, 61971004) (Corresponding author: Jungong Han).

Yunqi Miao and Xiao Ma are with the Warwick Manufacturing Group (WMG), University of Warwick, Coventry, CV4 7AL, United Kingdom (e-mail: Yunqi.Miao.1@warwick.ac.uk; X.Ma@warwick.ac.uk).

Zijia Lin is with Tsinghua University, Beijing, 100084, China (email: linzijia07@tsinghua.org.cn).

Guiguang Ding is with the School of Software, Tsinghua University, Beijing, 100084, China (e-mail: dinggg@tsinghua.edu.cn).

Jungong Han is with the Department of Computer Science, Aberystwyth University, Aberystwyth, SY23 3FL, United Kingdom (e-mail: juh22@aber.ac.uk).

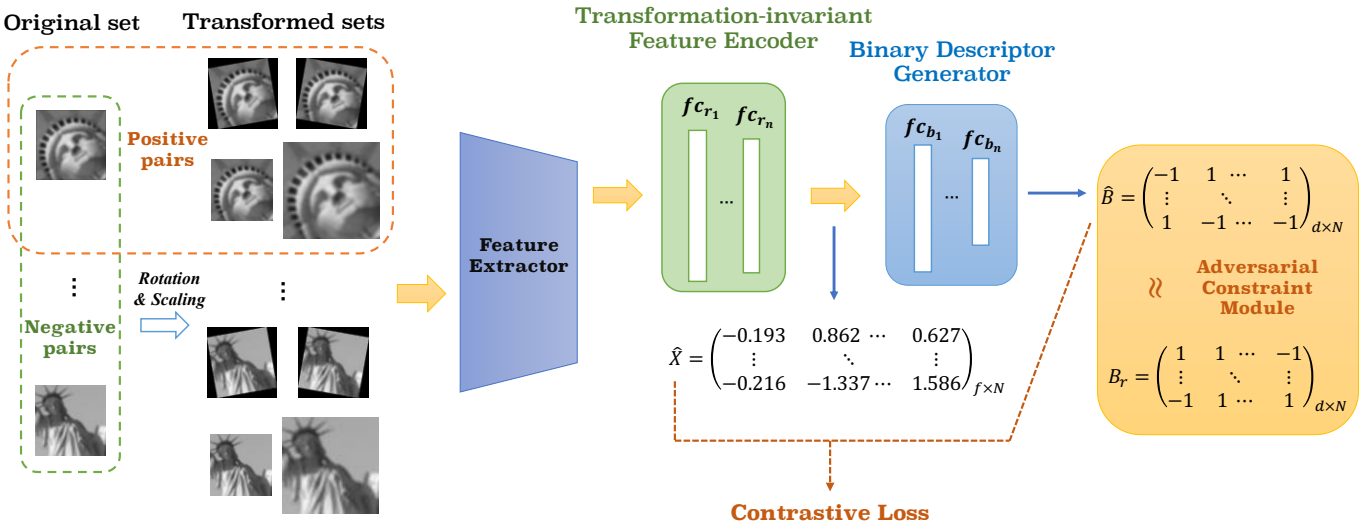


Fig. 1. The pipeline of the proposed TBLD. Firstly, patches from the "Original set" are augmented by rotating and scaling to build "Transformed sets". Then visual features of the image patches from the "Original set" and "Transform sets" are extracted from the VGG16 network. Subsequently, visual features are encoded by a *Transformation-invariant Feature Encoder* to obtain high-dimensional transformation-invariant features. On top of that, transformation-invariant binary local descriptors are obtained by *Binary Descriptor Generator*. f, d denote the dimension of high-dimensional transformation-invariant features and binary local descriptors, respectively. N is the number of training samples. Additionally, an *Adversarial Constraint Module* (ACM) is introduced to reduce bit correlations.

can be redundant during encoding. That would make the learned descriptor not compact enough. To explain the problem of correlated bits, we first evaluate the average amount of information conveyed by image patches and images with *Shannon entropy*. Then, we derive hash codes and binary local descriptors from images and patches, respectively, with two popular hashing methods: DeepBit [10] and Bi-half Net [18]. Later, we compare the correlations between bits under different code length settings with mean Absolute Correlations (mAC), which indicates the average correlation between bits. A higher mAC means a higher bit correlations. Details of mAC could be found in Section IV-D2. Specifically, the same number of images and image patches are randomly selected from an image dataset (CIFAR10 [19]) and an image patch dataset (Brown [20]) and are resized to the same size. The average Shannon entropy and mAC scores are illustrated in Table I. Seen from the results, the mAC scores under 32 bits and 64 bits settings are given by DeepBit since both source code and trained models are provided. The mAC scores under 128 bits and 256 bits settings are obtained by reproducing Bi-half Net based on the provided source code. Table I clearly demonstrates that images, with a higher average Shannon entropy, generally carry more complex information than image patches. When image hashing schemes are directly employed to derive binary local descriptors, the average correlations between bits exceed that of images by 1.17%, 3.22%, 9.9% and 10.13% under 32, 64, 128 and 256 bits settings in terms of mAC scores, respectively. Strong correlations between bits will undoubtedly deteriorate the representability of local descriptors [15], [21]. To mitigate the problem, most existing deep learning based works enforce the bits of binary local descriptors to be evenly distributed [10], [13], [22], which is performed on each training batch. However, such batch-based constraints generally suffer from a problem that the data distribution of a single batch cannot well represent that of the whole dataset due to the limited number of samples within a batch.

TABLE I
COMPARISON OF THE AVERAGE SHANNON ENTROPY BETWEEN IMAGES AND IMAGE PATCHES, AND THAT OF MEAN ABSOLUTE CORRELATIONS (MAC) (%) BETWEEN CORRESPONDING HASH CODES AND BINARY DESCRIPTORS, UNDER DIFFERENT CODE LENGTH SETTINGS.

	Entropy	mAC			
		32 bits	64 bits	128 bits	256 bits
Images	9.21	4.16	4.95	5.67	6.04
Patches	4.28	5.39	8.17	15.57	16.17

To tackle both limitations, in this paper, we propose a novel *Transformation-invariant Binary Local Descriptor* learning method (TBLD), which is trained in an unsupervised manner. The pipeline of TBLD is illustrated in Fig. 1. Specifically, it takes the "Original set" and "Transformed sets" as input. The former consists of the original image patches from the dataset, while the latter is built by rotating and scaling the original image patches. The framework aims to derive transformation-invariant and low-coupling binary local descriptors.

To generate transformation-invariant binary local descriptors, visual features extracted from original image patches and their transformed counterparts are enforced to be projected into an identical Euclidean subspace and an identical Hamming subspace simultaneously. Meanwhile, the distinctiveness between binary local descriptors of dissimilar image patches are maximized. To achieve that, instead of utilizing two separate terms during the optimization, an integrated loss term, the contrastive loss [23], is introduced here to propagate the neighboring structures of data from a high-dimensional feature space to a low-dimensional descriptor space. As a departure

from [23], where ALL the transformed samples within a training batch are employed as negative samples, we here propose a *Negative Pairs Selection Strategy* to adaptively select "Negative pairs" for each image patch during the training. By doing so, similar image patches from the same batch will form only ONE negative pair with a given image patch, instead of multiple negative pairs, thus dramatically reducing the computational costs. To the best of our knowledge, the contrastive loss is, for the first time, applied in the binary local descriptor learning work.

In the meantime, to reduce bit correlations, instead of manually imposing deterministic regularization terms on a batch of binary local descriptors, low-coupling binary codes are introduced externally here to guide the learning of binary local descriptors. Specifically, an *Adversarial Constraint Module* (ACM), which adopts the scheme of generator-discriminator, is adopted. The Wasserstein loss employed in the *Discriminator* minimizes the distributional discrepancy between the binary local descriptors generated by the framework and the introduced low-coupling binary codes. Although the proposed bottom-up learning strategy is employed at the batch level as the most correlations regularizers do, the optimization of *Discriminator* is an accumulated result of all previous batches, meaning that the adopted adversarial regularization is not restricted to the number of samples within a batch.

In summary, the contributions made in our work are mainly three-fold:

- An unsupervised binary local descriptor, which unites transformation-invariant and low-coupling properties, is proposed. To ensure the transformation invariance of binary local descriptors, the contrastive loss is, for the first time, applied in the learning of binary local descriptors. Instead of involving a large number of negative samples, a *Negative Pairs Selection Strategy* is proposed to selectively pick up a portion of "Negative pairs" for each training batch.
- We highlight the problem of the high correlations between bits in binary local descriptors when directly applying image hashing methods. To tackle that, we introduce a bottom-up learning strategy, termed *Adversarial Constraint Module* (ACM). Low-coupling binary codes generated externally are employed to guide the learning of binary local descriptors by minimizing their Wasserstein distances. This, by all means, is distinct from existing methods that simply using a hard threshold to enforce each bit to be evenly-distributive.
- Experimental results on three benchmark datasets show that our proposed descriptor surpasses existing binary descriptors by a clear margin in various visual tasks.

II. Related work

A. Binary Local Descriptors

1) Hand-crafted binary local descriptors: Binary local descriptors have attracted much attention due to their high matching efficiency over the past decade. Early binary local descriptors are typically hand-crafted and rely on intensity comparisons with a predefined pattern, like BRIEF [6], BRISK [5], and ORB [7], *etc*. These descriptors perform a set of pair-wise intensity comparisons within image patches to generate compact binary codes.

However, manually predefined sampling modes and intensity comparisons are sensitive to the geometric transformations and distortions on the original images, thereby leading to unstable performance.

2) Learning-based binary local descriptors: Later on, inspired by learning to hash methods for image retrieval [4], [24], learning-based binary local descriptors appeared [8], [9]. For instance, LDAHash [9] jointly minimizes the intra-class covariance of the descriptors and maximizes the inter-class covariance with Linear Discriminant Analysis (LDA), to produce a binary string from a SIFT descriptor. Binboost [8] aims to learn the illumination and viewpoint invariant binary descriptors with each bit being computed by a boosted binary hash function, which achieves state-of-the-art performance on patch matching task. However, these methods generally adopt simple binary intensity tests and thus are incapable of describing the domain-specific features of image patches [25].

Inspired by the advances in deep learning based image hashing approaches [22], [26], deep learning based binary descriptors have become dominant, which can be further categorized into supervised ones [12], [14], [15], [27] and unsupervised ones [10], [11], [13]. Supervised methods, which rely on the pair-wise/triplet-wise similarity labels of image patches to learn the descriptors, generally achieve better performance. L2-Net [12] is an end-to-end local descriptor learning framework, which preserves the neighboring relationship between matching pairs by enforcing the corresponding descriptors to be the nearest neighbors. On top of that, HardNet [14] maximizes the distance between matching pairs and the closest non-matching sample in a training batch, which extends the pair-wise mining strategy to triplet-wise one. CDbin [15] proposes a lightweight Convolutional Neural Network (CNN) to learn binary local descriptors, where the neighboring relationship of data is preserved and the bit information is enriched. Apart from learning from pair/triplet similarity labels, DOAP [27] proposes a novel list-wise learning-to-rank formulation for learning local feature descriptors, which directly optimizes a ranking-based retrieval performance metric, *i.e.,* Average Precision. Considering that annotated labels are expensive to obtain, supervised methods can probably be unfavorable in real-world applications.

Recently, unsupervised binary local descriptors [10], [11], [13], which do not require pair-wise similarity labels, have gained increasing attention. Existing works improve the representability of binary local descriptors from mainly two aspects: 1) enhancing the robustness to geometric transformations; 2) enriching the embedded information via reducing the bit correlations. For the former, DBD-MQ [11] enhances the quality of binary descriptors by applying a data-dependent binarization strategy. A K-AutoEncoders network is trained along with the holistic features to classify bits into the 0/1 categories with the minimal reconstruction error. Such a distribution strategy delivers stronger robustness, since bits from similar holistic features are more likely to be quantized into the same binary codes. Aside from improving the binarization

functions, GraphBit [13] improves the robustness of local descriptors by enhancing the responsibility of each bit. The mutual information between inputs and related bits are maximized, so that the ambiguous bits could receive additional instruction for confident binarization. DeepBit [10] augments patches via rotation and scaling, and employs a Siamese network to minimize the distances between the binary local descriptors of original image patches and their augmented counterparts. However, from the perspective of the essence of local descriptors, which is to describe the content in an image patch, we argue that the same content should be described by the *same* local descriptors in spite of viewpoints, instead of similar ones.

With respect to bit correlation reduction, existing works, *e.g.,* DH [22], DeepBit [10], GraphBit [13], UDBD [28], simply enforce the learned local descriptors to be evenly-distributive, *i.e.,* encouraging the mean of each bit to be 0.5 with the bit value ranging in $[0, 1]$. On top of that, BinGAN [16] embeds an adjusted Binarization Representation Entropy Regularizer to increase the entropy of the particular pairs of binary vectors that are not correlated in the high-dimensional feature space. Generally, such constraints are performed within training batches. However, the number of samples within a training batch is limited, meaning that the feature distribution of each batch cannot well represent that of the whole dataset. Therefore, imposing batch-based constraints typically fails to achieve the global optimum. Instead of performing constraints directly on the derived binary local descriptors, the framework here encourages to learn the mapping from the derived binary local descriptors to the low-coupling binary codes, which are introduced externally.

In the paper, the transformation invariance of binary local descriptors is achieved by projecting original image patches and their transformed counterparts into an identical Euclidean subspace and an identical Hamming subspace with the help of the contrastive loss. Additionally, we propose a bottom-up learning strategy assisted with Wasserstein loss to reduce bit correlations, where low-coupling binary codes are introduced externally to guide the learning of binary local descriptors.

B. GAN based Local Descriptor

Generative Adversarial Network (GAN) [29] has been extensively involved in unsupervised learning, where synthetic images are continuously generated to "fool" the network during training for improving the discriminability of the network. Inspired by its successful applications in feature learning [30], [31] and text-to-image generation [32], GAN has been recently introduced in the field of image hashing [33], [34]. HashGAN [33] utilizes generators to synthesize diverse images, and employs a discriminator to distinguish the synthetic images and the real ones. Meanwhile, a *Hash Encoder* learns the binary hash codes with the similarity information between images being preserved. BGAN [34] employs an auto-encoder to jointly learn binary hash codes in the middle and generate synthetic images at the end. The representability of the learned binary hash codes is improved by minimizing the distances between reconstructed images and the original ones. Meanwhile, the neighboring structures of images and features are also preserved. More recently, GAN has been applied in the learning-based binary descriptors [16]. BinGAN [16] takes an intermediate layer representation of a discriminator as the compact local binary descriptor. Two regularizers are also proposed to reduce the correlation between binary local descriptors.

Contrary to our work, HashGAN [33] and BGAN [34] are specifically designed for image retrieval task and use tanh-like activation for binarization. However, our work focuses on patch descriptor based tasks, like patch matching. Additionally, instead of taking the intermediate representations from the discriminator, we here employ *Discriminator* along with a set of low-coupling binary codes to guide the network to directly generate low-coupling binary local descriptors from *Descriptor Generator*.

III. Proposed method

A. Framework

To learn effective binary local descriptors, we propose a Transformation-invariant Binary Local Descriptor learning framework (TBLD), which improves the representability of local descriptors in terms of the robustness and bit correlations. To enable binary local descriptors to be invariant to transformations, inspired by visual representation learning [23], the contrastive loss is employed to preserve the neighboring structures of data. Specifically, the original image patches and their transformed counterparts are projected to an identical Euclidean subspace and an identical Hamming subspace, while the distinctiveness between binary local descriptors of dissimilar image patches are maximized. Additionally, an *Adversarial Constraint Module* (ACM) is introduced to reduce bit correlations, where low-coupling binary codes are introduced externally to guide the learning of binary local descriptors.

The pipeline of the proposed TBLD is depicted in Fig. 1. Specifically, given an image patch set $I^0 = \{I_i^0\}_{i=1}^c$ with c patches, I_i^0 refers to the i-th image patch. We firstly build v transformed patch sets, with each containing one certain type of transformation on the original image patches, like rotation or scaling. Then, the whole training set $I = \{I^i\}_{i=0}^v$ is formed by I^0 and the v transformed patch sets $\{I^i\}_{i=1}^v$. After that, visual features of all patches, $T = \{T^i \in \mathbb{R}^{t\times c}\}_{i=0}^v$, are extracted via the well-known VGG16 network [35], where t refers to the feature dimension. Subsequently, visual features are encoded by a *Transformation-invariant Feature Encoder* to obtain r-dimensional transformation-invariant features $\boldsymbol{X} \in \mathbb{R}^{r\times c}$. On top of that, a group of b-bit transformation-invariant binary local descriptors $\boldsymbol{B} \in \mathbb{R}^{b\times c}$ are obtained by binarizing the output of *Binary Descriptor Generator* $\boldsymbol{F} \in \mathbb{R}^{b\times c}$ as follows:

$$\boldsymbol{B} = sign(\boldsymbol{F}). \tag{1}$$

Here, we assume that $r{>}b$. As claimed, image patches and their transformed counterparts are united to the identical high-dimensional features $\boldsymbol{X}$ and binary local descriptors $\boldsymbol{B}$.

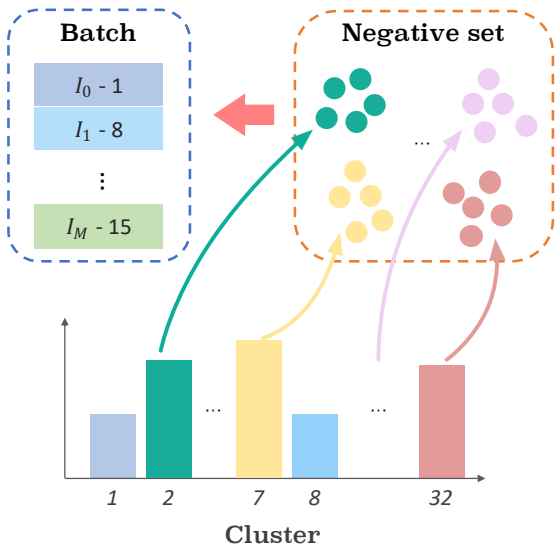


Fig. 2. The selection of "Negative sets" in *Negative Pairs Selection Strategy*. Visual features of image patches are firstly clustered into 32 clusters. For a given batch during training, the corresponding "Negative set" is formed according to the feature cluster of patches within the batch.

B. Transformation-invariant binary local descriptors

1) Selection of "Pseudo Positive pairs" and "Pseudo Negative pairs": To preserve the neighboring structures of image patches from the feature space to the descriptor space, the contrastive loss is performed after both *Transformation-invariant Feature Encoder* and *Binary Descriptor Generator*. For both modules, the feature representations of "Pseudo Positive pairs" are projected to an identical Euclidean subspace and an identical Hamming subspace. Meanwhile, the distinctiveness between the feature representations of "Pseudo Negative pairs" are maximized. Since pair-wise matching labels are not available here, we employ the neighboring relationships of image patches to build both "Pseudo Positive pairs" and "Pseudo Negative pairs". For simplicity, we use "Positive pairs" and "Negative pairs" to refer to "Pseudo Positive pairs" and "Pseudo Negative pairs", respectively.

Specifically, a "Positive pair" is built by an original patch and any one of its transformed counterparts. And a "Negative pair" is formed by an original patch and a sampled "Negative set". In our scenario, there are numerous image patches in the dataset, which means exhaustively pairing the given original patch with the rest of patches seems impractical in terms of the computational costs. Therefore, we propose a *Negative Pairs Selection Strategy*, which selectively picks up a "Negative set" to form the "Negative pairs". Concretely, given a batch with M image patches, q different image patches are selected to form a "Negative set" according to their *"clusters"*. The selection of "Negative set" is illustrated in Fig. 2, where the extracted visual features T are clustered into 32 clusters offline. During training, for a given batch, the cluster distribution of the image patches within the batch is analyzed. Then samples are randomly selected from the uncovered clusters to build the "Negative set". If all the clusters are covered, samples are selected randomly and evenly from each cluster to form the "Negative set". In the experiment, q is empirically set as 4096, considering the balance between computational cost and data diversity.

2) Contrastive loss: Given the "Positive pairs" and "Negative pairs", the contrastive loss is performed after both *Transformation-invariant Feature Encoder* and *Binary Descriptor Generator* to propagate the neighboring structures from high-dimensional features to compact binary local descriptors. The two loss terms are represented by L_{C_r} and L_{C_b}, respectively, which enforce "Positive pairs" to have identical transformation-invariant high-dimensional features and compact binary local descriptors, respectively. Meanwhile, the distinctiveness between feature representations of "Negative pairs" is maximized. Firstly, L_{C_r} is formulated as follows.

$$L_{C_r} = -\sum_{i=0}^{v}\sum_{m=1}^{M}\frac{\alpha_i^{\gamma}}{M} log\frac{e^{-s_r Dist_E(x_m^i, x_m)}}{e^{-s_r \overline{Dist_E}(x_m^i, x_{neg})}}, \tag{2}$$

where x_m^i denotes the output of *Transformation-invariant Feature Encoder* of the m-th patch from the i-th "Transformed sets", and x_m is the transformation-invariant high-dimensional feature of the m-th patch. $Dist_E(x_m^i, x_m)$ represents the distance between x_m^i and x_m, which is formulated as follows,

$$Dist_E(x_m^i, x_m) = \|x_m^i - x_m\|_2^2. \tag{3}$$

In the denominator, we adopt the average Euclidean distance between x_m^i and the corresponding high-dimensional *"negative"* set x_{neg} formed by the real-valued representations of q *"negative"* samples. which is denoted as,

$$\overline{Dist_E}(x_m^i, x_{neg}) = \frac{1}{q}\sum_{x_j \in x_{neg}} \|x_m^i - x_j\|_2^2. \tag{4}$$

In Eq. (2), α_i denotes the to-be-learned non-negative weight w.r.t the i-th "Transformed sets", which sums up to 1. γ is a smoothing parameter and s_r denotes the temperature parameter for real-valued local descriptors, which are empirically set as 3 and 0.1, respectively.

Similarly, the contrastive loss applied after *Binary Descriptor Generator*, *i.e.,* L_{C_b}, is defined as,

$$L_{C_b} = -\sum_{i=0}^{v}\sum_{m=1}^{M}\frac{\alpha_i^{\gamma}}{M} log\frac{e^{-s_b Dist_H(b_m^i, b_m)}}{e^{-s_b \overline{Dist_H}(b_m^i, b_{neg})}}, \tag{5}$$

where b_m^i denotes the binary feature representations of the m-th patch in the i-th "Transformed sets", and b_m indicates the transformation-invariant binary local descriptor of the m-th patch. $Dist_H$ denotes the Hamming distance and $\overline{Dist_H}$ represents the average Hamming distance between the binary string of the given image patch and the binary local descriptors of its counterparts from the "Negative set" b_{neg}. s_b denotes the temperature parameter for binary local descriptors, which is empirically set as 0.1.

However, directly optimizing binary values will make the back-propagation of the framework infeasible, which is known as the *ill-posed gradient* problem [34]. In the paper, we replace b_m^i and b_m in Eq. (5) with the relaxed real-valued representations output by *Binary Descriptor Generator* before

the binarization f_m^i, and the to-be-binarized transformation-invariant local descriptors f_m, respectively. Then Eq. (5) can be rewritten as follows.

$$L_{C_b} = -\sum_{i=0}^{v}\sum_{m=1}^{M}\frac{\alpha_i^\gamma}{M} log\frac{e^{-s_b Dist_E(f_m^i, f_m)}}{e^{-s_b \overline{Dist_E}(f_m^i, f_{neg})}}. \quad (6)$$

The replacement requires a low quantization error between the binary local descriptors and the corresponding relaxed real-valued feature representations. Therefore, a quantization loss term L_Q is employed, which is denoted as,

$$L_Q = \sum_{i=0}^{v}\sum_{m=1}^{M}\frac{\alpha_i^\gamma}{M}\|f_m^i - b_m\|_2^2, \quad (7)$$

where f_m^i denotes the relaxed real-valued feature representations of the m-th patch in the i-th "Transformed sets". And b_m refers to the transformation-invariant binary local descriptor of the m-th patch, which refers to the m-th column in $\boldsymbol{B}$.

C. Low-coupling binary local descriptors

Apart from enhancing the robustness of binary local descriptors against transformations, decorrelating bits of the compact descriptor is also of great importance. As revealed previously, correlated bits convey overlapped information, thus weakening the representation capacity of the binary local descriptors. According to [36], *Wasserstein distance* can measure the distance between two non-overlapped data distributions, which perfectly fits the situation where discrete and continuous distributions coexists. Inspired by this, we advocate the use of Wasserstein loss to minimize the *Wasserstein distance* between the data distribution of low-coupling binary codes and the feature distribution of the derived binary local descriptors. Although the Wasserstein loss has been successfully employed in applications like person re-identification [37], [38], it has never been employed to learn binary local descriptors yet.

In the paper, a bottom-up learning strategy is proposed to reduce bit correlations, termed *Adversarial Constraint Module* (ACM). The structure of ACM is depicted in Fig. 3, which adopts the scheme of generator-discriminator in the adversarial learning.

Specifically, the proposed framework serves as a *Descriptor Generator* to derive binary local descriptors. Meanwhile, a sampler is employed to generate low-coupling binary codes by *randomly* and *independently* sampling 0/1 values from the Bernoulli distribution with the probability $p = 0.5$, which conforms to the principle of local descriptors in [10]. Given the input, a *Discriminator*, consisting of 3 fully-connected (fc) layers, is followed. The first two fc layers are followed by a ReLU activation function. In *Discriminator*, the Wasserstein loss is employed to encourage the derived binary local descriptors to mimic the distribution of the low-coupling binary codes by alternately optimizing the *Discriminator* and the *Descriptor Generator*.

Formally, given a training batch with M image patches $I = \{I_i\}_{i=0}^M$, a batch of binary local descriptors $\boldsymbol{B} = \{b_i\}_{i=0}^M$ could be learned by the *Descriptor Generator*. Similarly, to avoid the *ill-posed gradient* problem, we replace the binary

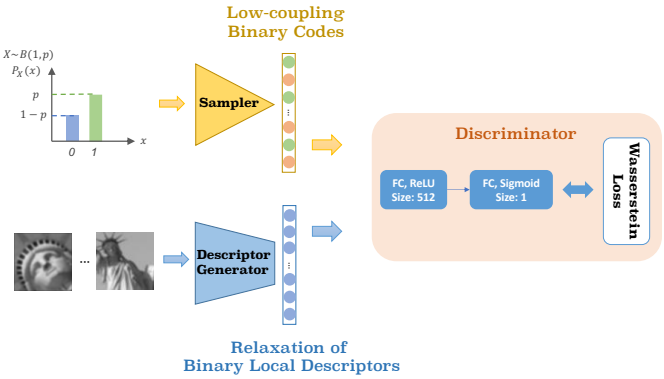


Fig. 3. The structure of *Adversarial Constraint Module* (ACM). The *Discriminator* takes the derived binary local descriptors from the *Descriptor Generator* and the low-coupling binary codes sampled from the Bernoulli distribution as input. With the help of Wasserstein loss, *Discriminator* and *Descriptor Generator* are alternately trained to learn the mapping from the derived binary local descriptors to the low-coupling binary codes.

local descriptors $\boldsymbol{B}$ with the relaxed representations $\boldsymbol{F} = \{f_i\}_{i=0}^M$, which refers to the real-valued feature representations output by the *Binary Descriptor Generator* before binarization. Additionally, the low-coupling binary codes $\boldsymbol{B_r} = \{b_{r_i}\}_{i=0}^M$ are sampled under the Bernoulli distribution. The *Wasserstein distance* between $\boldsymbol{F}$ and $\boldsymbol{B_r}$ could be approximated by,

$$\max \mathbb{E}_{b_{r_i}\sim b}[D(b_{r_i})] - \mathbb{E}_{f_i\sim\mathbb{P}_f}[D(f_i)], \quad (8)$$

where D refers to the discriminator. $\mathbb{P}_f$ and b refer to the feature distribution of $\boldsymbol{F}$ and data distribution of $\boldsymbol{B_r}$, respectively.

According to [36], Eq. (8) holds only when *Lipschitz constraint* is satisfied. Therefore, following [39], *Lipschitz constraint* is enforced by penalizing the p-norm of the gradient of the discriminator w.r.t. the input, *i.e.,* $\|\nabla_x D(x)\|_p \leq 1$. According to [39], enforcing the gradient norm constraint everywhere is intractable, so we only enforce it on the space that is uniformly sampled from the feature distribution $\mathbb{P}_f$ and the data distribution b. Integrating the regularizer to the objective function, Wasserstein loss employed in *Adversarial Constraint Module* can be denoted as follows.

$$\begin{aligned} L_W = &-\mathbb{E}_{b_{r_i}\sim b}[D(b_{r_i})] + \mathbb{E}_{f_i\sim\mathbb{P}_f}[D(f_i)] \\ &+\eta\mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2], \end{aligned} \quad (9)$$

where $\hat{x} \sim \mathbb{P}_{\hat{x}}$ is sampled from both inputs with a random sample weight $\epsilon \sim U[0,1]$, and it can be formulated as,

$$\hat{x} = \epsilon f_i + (1-\epsilon) b_{r_i}. \quad (10)$$

Notably, there are some negative values in the $\boldsymbol{F}$ since the *Binary Descriptor Generator* is trained to push $\boldsymbol{F}$ to $[-1, 1]$. However, the sampled binary local descriptors $\boldsymbol{B_r} \in [0, 1]$. To unify the two inputs, we replace the 0 in sampled binary local descriptors $\boldsymbol{B_r}$ with -1. Additionally, to eliminate the input noise, we also apply L_2 normalization on $\boldsymbol{F}$ before sending it to the Discriminator.

D. Loss Function

As our method adopts the scheme of generator-discriminator, two learning objectives, *i.e.,* L_G for the *Descrip-*

tor Generator and L_D for the *Discriminator*, are employed, respectively.

1) Descriptor Generator Objective: Given the 1) the contrastive loss L_{C_r} in Euclidean space, 2) the contrastive loss L_{C_b} in Hamming space, and 3) quantization loss L_Q, the objective for *Descriptor Generator* is written as follows:

$$L_G = L_{C_r} + L_{C_b} + \beta L_Q - \lambda_D \mathbb{E}_{f_i \sim \mathbb{P}_f}[D(f_i)], \tag{11}$$

where β balances the contribution of L_Q, and λ_D controls the penalty of the *Discriminator*, which are both empirically set as 1. Note that, to avoid plunging the network into the trivial solution, where all the real-valued feature representations become an all-zero or infinite matrix, we enforce the L_2-norm of the real-valued feature representations of each query to be 1, *i.e.,* $\|x_m^i\|_2 = 1$. To simplify the learning process, we integrate the constraint in the objective as L_N, which is denoted as follow.

$$L_N = \sum_{i=0}^{v} \sum_{m=1}^{M} \frac{\alpha_i^\gamma}{M} (1 - \|x_m^i\|_2), \tag{12}$$

Therefore, the objective L_G can be formulated as,

$$L_G = L_{C_r} + L_{C_b} + \beta L_Q + \lambda L_N - \lambda_D \mathbb{E}_{f_i \sim \mathbb{P}_f}[D(f_i)], \tag{13}$$

where λ is the weight for the regularizer L_N, which is set as 1e-5 empirically.

2) Discriminator Objective: The *Discriminator* objective L_D is defined by,

$$L_D = \lambda_D L_W, \tag{14}$$

where λ_D is the same hyper-parameter as in Eq. (13). In the paper, *Descriptor Generator* and *Discriminator* are trained with the SGD [40] optimizer with the initial learning rate being 5e-7 and 1e-8, respectively.

E. Optimization

The training procedure of TBLD is summarized in Algorithm 1. Firstly, the parameters of *Descriptor Generator*, w, are initialized following the Kaiming initialization [41]. The non-negative weights for the "Original set" and "Transformed sets", $\alpha = \{\alpha_i\}_{i=0}^v$, are all initialized as $1/v$. Given the to-be-learned variables, *i.e.,* the transformation-invariant high-dimensional features $\boldsymbol{X}$, the transformation-invariant binary local descriptors $\boldsymbol{B}$, and weights for input α, an alternating optimization method is proposed to solve the objective Eq. (13) via conducting the following steps iteratively.

(i) **Update $\boldsymbol{X}$.** With B, w_r, w_b, α fixed, the objective function w.r.t. $\boldsymbol{X}$ can be rewritten as,

$$\psi_1 = \min_X L_{C_r}. \tag{15}$$

By setting the derivation of Eq. (15) w.r.t. $\boldsymbol{X}$ as 0, we can get the closed-form solution of $\boldsymbol{X}$:

$$\boldsymbol{X} = \frac{\sum_{i=1}^v \alpha_i^\gamma \boldsymbol{X}^i}{\sum_{i=1}^v \alpha_i^\gamma}. \tag{16}$$

(ii) **Update $\boldsymbol{B}$.** Similarly, with other parameters fixed, we can rewrite the objective function w.r.t. $\boldsymbol{B}$ as follows.

$$\psi_2 = \min_B L_Q. \tag{17}$$

Algorithm 1 The training procedure of the proposed TBLD.

Input: Number of training batch in one epoch, N_b; Batch size, M;
Output: Binary local descriptors, $\boldsymbol{B}$;

```
Initialize Descriptor Generator, Discriminator, and α_i;
Initialize X_0 and B_0 by Eq. (16) and Eq. (18);
repeat
    for i = 0 → N_b do
        Sampling ε ~ U[0, 1];
        Sampling B_r ~ b(M, 0.5);
        Deriving X^i, F^i, B^i by Descriptor Generator;
        Optimizing Discriminator according to Eq. (14);
        Optimizing Descriptor Generator according to Eq. (13);
    end for
    Updating α_i by Eq. (20).
    Updating X and B with the updated Descriptor Generator.
until convergence.
```

According to Eq. (1), $\boldsymbol{B}$ can be obtained by binarizing $\boldsymbol{F}$ with the *sign* function. $\boldsymbol{F}$ can be obtained in a similar manner with $\boldsymbol{X}$. To conclude, $\boldsymbol{B}$ can be obtained as,

$$\boldsymbol{B} = sign\Big(\frac{\sum_{i=1}^v \alpha_i^\gamma \boldsymbol{F}^i}{\sum_{i=1}^v \alpha_i^\gamma}\Big). \tag{18}$$

(iii) **Update α.** At the end of each epoch, with other parameters fixed, we can rewrite the objective function w.r.t. α as follows.

$$\psi_3 = \min_\alpha \sum_{i=0}^{v} \alpha_i^\gamma (L_{C_r}^{\tilde{i}} + L_{C_b}^{\tilde{i}} + \beta L_Q^{\tilde{i}} + \lambda L_N^{\tilde{i}}), \tag{19}$$

where $L_{C_r}^{\tilde{i}}$, $L_{C_b}^{\tilde{i}}$, $L_Q^{\tilde{i}}$, $L_N^{\tilde{i}}$ are obtained from L_{C_r}, L_{C_b}, L_Q, L_N by factoring out α_i^γ, respectively. Suppose that $L^i = L_{C_r}^{\tilde{i}} + L_{C_b}^{\tilde{i}} + \beta L_Q^{\tilde{i}} + \lambda L_N^{\tilde{i}}$, the optimal α can be derived as,

$$\alpha_i = \frac{(L^i)^{\frac{1}{1-\gamma}}}{\sum_{j=0}^v (L^j)^{\frac{1}{1-\gamma}}}. \tag{20}$$

After alternately updating the parameters in *Descriptor Generator* and *Discriminator* until the network converges, the low-coupling binary local descriptors are derived from the *Descriptor Generator*.

IV. EXPERIMENT

We evaluate the proposed binary local descriptor learning method on three widely used public datasets, *i.e.,* **Brown** [20], **HPatches** [43], and **Mikolajczyk** [44]. Comparisons with the state-of-the-arts are conducted on visual analysis tasks like patch matching, patch retrieval, and patch verification. In this section, we will start by introducing the datasets and experimental settings, then present and analyze the comparison results.

TABLE II
COMPARISONS OF 95% ERROR RATE (%) WITH THE STATE-OF-THE-ART LOCAL DESCRIPTORS ON THE BROWN DATASET. THE CODE LENGTHS ARE INDICATED BY DIM AND BYTES FOR REAL-VALUED LOCAL DESCRIPTORS AND BINARY ONES, RESPECTIVELY.

	Train	*Yosemite*		*Notre Dame*		*Liberty*		Average
	Test	*Notre Dame*	*Liberty*	*Yosemite*	*Liberty*	*Notre Dame*	*Yosemite*	Err
Real-valued	SIFT [17] (128 dim)	28.09	36.27	29.15	36.27	28.09	29.15	31.17
Binary (Supervised)	D-BRIEF [42] (4 bytes)	43.96	53.39	46.22	51.30	43.10	47.29	47.54
	BinBoost [8] (8 bytes)	14.54	21.67	18.96	20.49	16.90	22.88	19.24
Binary (Unsupervised)	BRIEF [6] (32 bytes)	54.57	59.15	54.96	59.15	54.57	54.96	56.23
	BRISK [5] (64 bytes)	74.88	79.36	73.21	79.36	74.88	73.21	75.81
	ORB [7] (32 bytes)	48.03	56.26	54.13	56.26	48.03	54.13	52.81
	DeepBit [10] (32 bytes)	28.49	34.64	54.63	33.83	20.66	56.69	38.15
	DBD-MQ [11] (32 bytes)	20.13	25.77	50.99	22.92	18.95	50.36	31.52
	BinGAN [16] (32 bytes)	16.88	26.08	40.80	25.76	27.84	47.64	30.76
	GraphBit [13] (32 bytes)	17.78	24.72	49.94	21.18	15.25	49.64	29.75
	TBLD (32 bytes)	**16.53**	**21.95**	**35.09**	**20.45**	**14.47**	**36.88**	**18.25**

A. Dataset Descriptions

- ***Brown dataset*** [20] contains three subsets: *Liberty*, *Notre Dame* and *Yosemite*. Each subset contains 400,000 to 600,000 gray-scale image patches for training and 100,000 patch pairs for testing. The size of image patches in the dataset is 64×64. For the test sets, half of the pairs are matched and others are non-matched. In the experiment, we follow the settings in [8], *i.e.,* training the network with one subset and then evaluating it on the other two subsets. There are 6 train-test combinations in total.
- ***HPatches dataset*** [43] consists of 116 image sequences, with 59 containing significant viewpoint changes and the rest containing illumination deformations. Each sequence includes a reference image and 5 target images. Image patches are detected in the reference image with Difference of Gaussians (DoG) detector and projected on the target images using the ground truth homographies. The sizes of patches are normalized to 65×65. Following the setting in [15], the to-be-evaluated model in this experiment is trained on the *Liberty* subset of the **Brown** dataset.
- ***Mikolajczyk dataset*** [44] is proposed to investigate the robustness of descriptors to viewpoints (*Graffiti*), compression artifacts (*Ubc*), illumination changes (*Leuven*), blurriness (*Trees*), and zoom and rotation (*Boat*). Each subset comprises a reference image and 5 target images, which are sorted by an increasing degree of distortions. Since TBLD is proposed to deal with the scale and viewpoint transformation, the evaluation is conducted on the corresponding scenes, *i.e., Boat* and *Graffiti*. Following the protocol in [44], SIFT keypoint detector is firstly employed to detect 1000 interest points for each image in an image pair. Then the keypoints are matched via an exhaustive search based on the Hamming distance between the corresponding binary descriptors. Following the setting in [8], the to-be-evaluated model in this experiment is trained on the *Notre Dame* subset of the **Brown** dataset.

B. Implementation Details

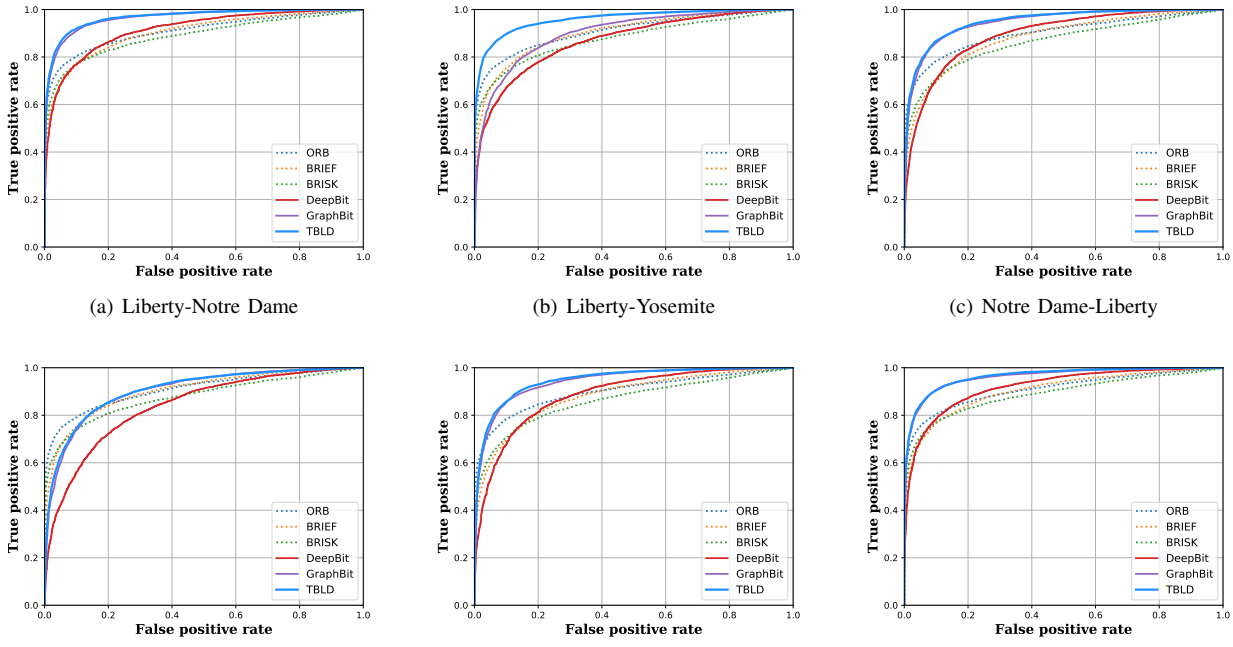
To preprocess the input data, two types of transformations, *i.e.,* rotation and scaling, are employed to derive the transformed sets. Rotation angles range in $\{-10, -5, 5, 10\}$, and scaling factors are set as 0.8 and 1.2. To obtain features from the *fc7* layer (4096-d) of the pre-trained VGG16 [35], the input patches are firstly resized into 256×256 and then cropped to 224×224. We here set the length of real-valued and binary local descriptors as 1024 and 256, following the setting of [13]. The batch size is 32 and the maximum iteration is 10000.

C. Comparison With State-of-the-Arts

1) Results on Brown Dataset: Experiments on the Brown dataset aim to evaluate the performance of the proposed approach on the patch matching task. Following [10], [11], [13], the adopted evaluation metric is 95% error rate, which denotes the percent of incorrect matches when 95% of the ground-truth matched patches are found. *Lower 95% error rate represents better performance*. Comparisons are conducted with the state-of-the-art works, including supervised descriptors (*e.g.,* D-BRIEF [42], and BinBoost [8]) and unsupervised ones (*e.g.,* BRISK [5], BRIEF [6], and GraphBit [13], *etc.*). The comparison results are reported in Table II, where the results of real-valued descriptor SIFT [17] and supervised descriptors are also provided as references.

As can be seen, TBLD outperforms the state-of-the-art unsupervised binary descriptors, including both hand-crafted and deep learning based ones, on all the subsets. A decline of 11% can be found in terms of 95% error rate in contrast to the best unsupervised binary local descriptor learning method so far (GraphBit [13]). It is worth mentioning that when compared with a widely-used floating-point descriptor, *i.e.,* SIFT [17], the proposed TBLD obtains a lower 95% error rate, along with a much lower computation cost for measuring similarities.

Moreover, Receiver Operating Characteristic (ROC) curves of the state-of-the-art unsupervised binary local descriptors are plotted in Fig. 4. The curves illustrate the true positive rate (TPR) against false positive rate (FPR) at various threshold settings. For a fair comparison, we firstly reproduce the algorithms and then plot the curves. In terms of deep learning based binary local descriptors, the competitors include DeepBit [10] and GraphBit [13] because only their source codes are provided and GraphBit [13] still maintains the best performance until now. As can be seen, the ROC curves from TBLD rank at the top on all train-test configurations.



(a) Liberty-Notre Dame (b) Liberty-Yosemite (c) Notre Dame-Liberty

(d) Notre Dame-Yosemite (e) Yosemite-Liberty (f) Yosemite-Notre Dame

Fig. 4. ROC curves of the proposed TBLD and the state-of-the-art on the Brown dataset, with all the train-test combinations among three subsets.

2) Results on HPatches Dataset: We use HPatches dataset to evaluate the performance of binary local descriptors on three visual tasks: patch matching, patch retrieval, and patch verification. Specifically, descriptors are compared in the matching task to find matched patches between the reference image and the target ones. For the patch retrieval task, local descriptors are employed to match a query patch to a pool of patches extracted from many images. In terms of patch verification, descriptors are utilized to classify whether two patches are matched or not.

Following the evaluation metrics suggested by [43], we compare TBLD with the state-of-the-art descriptors in terms of mean average precision (mAP). The comparison results are reported in Table III. *Higher mAP means better performance.* Again, the binary local descriptors are categorized as supervised and unsupervised ones according to the training manner. Since DBD-MQ [11] and BinGAN [16] did not report the results on the HPatches dataset, they are not included in Table III.

It can be seen that TBLD beats all unsupervised baselines, including both hand-crafted ones (BRISK [5], BRIEF [6], ORB [7]) and deep learning based ones (DeepBit [10], GraphBit [13]) on all the tasks. Specifically, compared to GraphBit [13], TBLD improves the mAP score by 8.2%, 6.8%, 4.5%, respectively, in the three tasks. Here we also provide the result of the real-valued SIFT [17] for a reference. It can be observed that our method even outperforms SIFT on the patch verification task with a 4.58 % increase in terms of mAP.

3) Results on Mikolajczyk Dataset: Experiments are conducted on the *Mikolajczyk* dataset [44] to prove the generalization of the binary local descriptors on the patch matching task. Here, we compare TBLD with both hand-crafted binary local descriptors (BRISK [5], ORB [7]), and the best deep learning based one so far (GraphBit [13]). Considering the fairness, the binary local descriptors are set as 32 bytes for all the methods. Specifically, we firstly reproduce the algorithm and then employ the *Recognition rate* to evaluate the performance following [6], [17]. The *Recognition rate* can be obtained as follows,

- Extracting n_1 interest points from the reference image, and n_2 from the target image. Among them, n matching pairs are obtained from the ground-truth homograph transformation matrix.
- For each interest point in the reference point set, finding the nearest neighbor in the target point set via binary local descriptors.
- Counting the number of correct matches n_c, and calculating the recognition rate with $r = n_c/n$.

Following the previous works [6], [8], for an image, interest points are firstly detected by the SURF Hessian-based detector and patches are then cropped and normalized to the required size of each descriptor. Specifically, for BRISK [5] and ORB [7], the sizes of patches keep unchanged. As for GraphBit and

TABLE III
COMPARISONS OF MAP (%) WITH THE STATE-OF-THE-ART BINARY LOCAL DESCRIPTORS ON THE HPATCHES DATASET.

	Method	Match	Retrieval	Verification
Real-valued	SIFT [17] (128 dim)	25.47	31.98	65.12
Binary (Supervised)	BinBoost [8] (32 bytes)	16.97	38.68	76.27
Binary (Unsupervised)	BRIEF [6] (32 bytes)	10.50	16.03	58.07
	BRISK [5] (64 bytes)	15.97	18.10	65.65
	ORB [7] (32 bytes)	15.32	18.85	60.15
	DeepBit [10] (32 bytes)	13.05	20.61	61.27
	GraphBit [13] (32 bytes)	14.22	25.19	65.19
	TBLD (32 bytes)	**15.39**	**27.03**	**68.25**

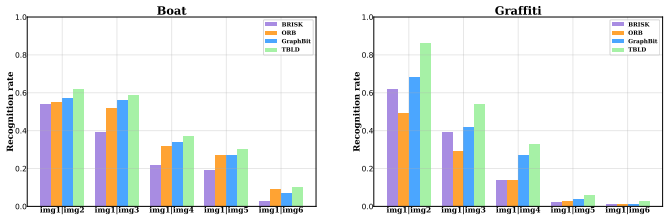


Fig. 5. Recognition rate on the Mikolajczyk dataset. TBLD outperforms other state-of-the-art binary local descriptors learning approaches in terms of recognition rate on all reference-target configurations.

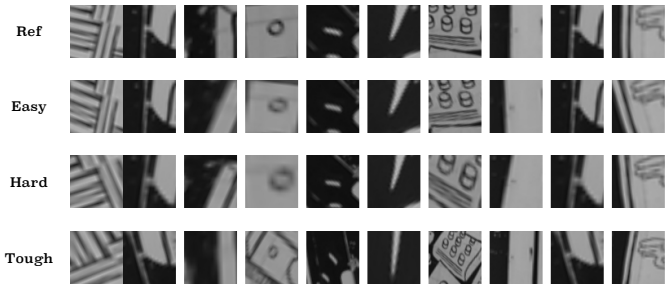


Fig. 6. Visualization of patches from the Hpatches dataset. Patches from the reference image (REF) are shown in the first row. Patches from target images with the increasing level of geometric noises: EASY, HARD and TOUGH, are shown in row 2 to 4, respectively.

our method, the patches are resized to 224×224 for feature extraction.

Fig. 5 illustrates the recognition rates of the state-of-the-art binary local descriptors on both *Boat* and *Graffiti* scenes with the challenges of zoom/rotation and viewpoint variations, respectively. As can be seen, TBLD outperforms other state-of-the-art binary local descriptors on all the reference-target configurations. Additionally, it can be found that, compared to the *Boat* scene, our method performs better on the *Graffiti* scene, where a significant increase in the recognition rate can be seen for all configurations. We attribute it to that the gap between training scenes (*Notre Dame*) and *Graffiti* is relatively smaller than *Boat*. Results on the *Mikolajczyk* dataset further verify the generalization ability of the proposed method.

4) Comparison using a unified metrics: Additionally, to show the superiority of our method clearly, we evaluate the performance of the learned binary descriptors on the patch matching task of the three datasets in terms of a unified metrics: mAP. For fair comparisons, we have to reproduce some representative baselines, including hand-crafted ones (BRIEF and ORB) and deep learning based one (GraphBit), and evaluate their performances by ourselves, because existing algorithms failed to evaluate their works using the same metric on different datasets. Due to time constraints, only state-of-the-art binary descriptor learning methods are chosen.

The mAP scores are illustrated in Table IV. The model used for evaluation is trained on the Liberty subset of the Brown dataset. Note that, the mAP scores of the Mikolajczyk dataset are obtained by the patch matching between the target image (img1) and the reference image with the mildest distortion (img2). As can be seen, the proposed TBLD still outperforms the state-of-the-art approaches on the three datasets when a unified metric is employed.

TABLE IV
COMPARISON OF MAP (%) WITH THE STATE-OF-THE-ART BINARY DESCRIPTORS ON THE THREE DATASETS.

Method	Brown (Liberty)		Hpatches	Mikolajczyk	
	Notre Dame	Yosemite		Boat	Graffiti
BRIEF	62.05	66.40	16.03	43.24	34.34
ORB	64.19	68.63	18.85	51.11	44.83
GraphBit	68.78	72.27	25.19	59.19	57.39
TBLD	**69.52**	**74.39**	**27.03**	**62.41**	**60.07**

D. Ablation Study

1) Transformation Invariance: Firstly, we investigate the transformation invariance of binary local descriptors derived by TBLD. Since the to-be-evaluated models are trained on the Brown dataset, which probably tends to adapt better to the transformations within the dataset, we conduct the analysis on the other two datasets. For the Mikolajczyk dataset, as discussed above, both *Boat* and *Graffiti* subsets contain a certain type of transformations, which means the transformation invariance of the derived binary local descriptors has been proved by the results in Fig. 5. Therefore, here the ablation study is conducted on the HPatches dataset [43] to evaluate the robustness of the derived binary local descriptors against geometric noises and viewpoints.

Geometric noise. Specifically, image patches in the HPatches dataset have been divided into different subsets according to the level of geometric noises, which are indicated by EASY, HARD and TOUGH. Examples of the reference and the target image patches in each subset are shown in Fig. 6. On each subset, we compare our TBLD with other transformation-invariant binary local descriptors: BRISK [5], ORB [7] and DeepBit [10] in terms of mAP, following the settings in [43]. The results are illustrated in Fig. 7. As can be seen, the proposed TBLD achieves a higher mAP on all the subsets, which proves the robustness of our method against multiple levels of geometric noises.

Viewpoints. For the task of patch matching, HPatches further groups the data with different levels of geometric noises, into "ILLUM" and "VIEW" subsets, to facilitate the evaluation of the robustness of binary local descriptors against illumination and viewpoints changes. Since the proposed TBLD focuses on improving the robustness of descriptors against transformations like rotation and scaling, the comparison is specifically conducted on the "VIEW" subsets. The mAP of the state-of-the-art unsupervised transformation-invariant binary local descriptors on the "VIEW" subsets are illustrated in Fig. 8.

As can be seen, although our method underperforms BRISK [5] on the overall mAP of the patch matching task (15.39% and 15.97%, respectively), it outperforms BRISK on the three "VIEW" subsets from EASY, HARD, and TOUGH, respectively. The results prove the robustness of the binary local descriptors derived by TBLD against viewpoints changes.

2) Effectiveness of Adversarial Constraint Module: We prove the effectiveness of the *Adversarial Constraint Module* (ACM) from two aspects. 1) We compare the proposed

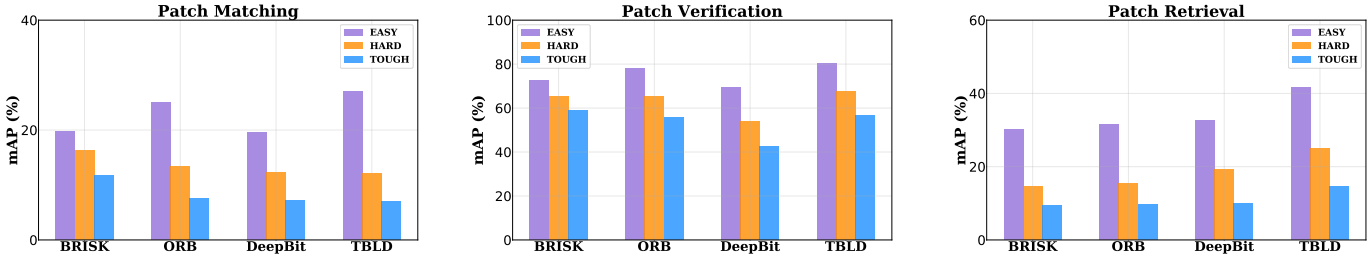


Fig. 7. Performance of the state-of-the-art transformation-invariant binary local descriptors on the three tasks in the HPatches dataset. For each task, data are divided into three subsets according to the level of geometrical noises, which are indicated by EASY, HARD, and TOUGH, respectively.

TABLE V
COMPARISON OF 95% ERROR RATE (%) WITH THE MODEL TRAINED WITH THE EVENLY-DISTRIBUTIVE CONSTRAINT ON THE BROWN DATASET.

Train	*Yosemite*		*Notre Dame*		*Liberty*	
Test	*Notre Dame*	*Liberty*	*Yosemite*	*Liberty*	*Notre Dame*	*Yosemite*
Evenly-distributive constraint	18.16	24.10	36.57	21.93	15.72	37.67
ACM	**16.53**	**21.95**	**35.09**	**20.45**	**14.47**	**36.88**

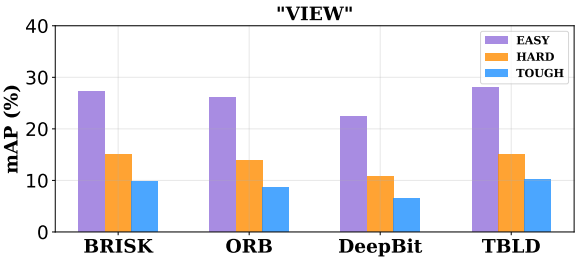


Fig. 8. Comparison with the state-of-the-art transformation-invariant binary local descriptors on the "VIEW" subsets of the patch matching task in terms of mAP(%).

bottom-up strategy (ACM) with the evenly-distributive constraint employed to reduce the bit correlations in the existing works, such as DH [22] and DeepBit [10]. 2) We investigate the contribution of ACM by comparing the bit correlations of binary descriptors derived by models trained with (w/) and without (w/o) ACM. Since the HPatches [43] and the Mikolajczyk [44] are designed only for evaluation, we only conduct the comparisons on the Brown dataset [20].

Comparison with the evenly-distributive constraint. To compare with the evenly-distributive constraint, we remove ACM from the framework and employ the evenly-distributive constraint, which is denoted as follows,

$$L_M = \sum_{k=1}^{K} ||\mu_k - 0,5||^2, \qquad \mu_k = \frac{1}{N}\sum_{n=1}^{N} b_{nk}, \tag{21}$$

where K is the length of binary descriptors and μ_k denotes the mean value of each bit over N samples within a mini-batch. Therefore, the overall objective for the to-be-compared models can be derived as,

$$L = L_{C_r} + LC_b + \beta L_Q + \lambda L_N + L_M. \tag{22}$$

The performances of the derived binary descriptors are evaluated in terms of 95% error rates, which are reported in Table V. It can be observed that binary descriptors derived by the proposed method outperform those derived by the model trained with the evenly-distributive constraint with lower 95% error rates.

TABLE VI
COMPARISON OF THE MEAN ABSOLUTE CORRELATIONS (%) OF BINARY LOCAL DESCRIPTORS DERIVED BY MODELS WITH (W/) AND WITHOUT (W/O) *Adversarial Constraint Module*. Δ REFERS TO THE RELATIVE DECREASE.

	w/	**w/o**	Δ
Yosemite	13.73	14.25	-3.65%
Notre Dame	7.43	9.64	-22.92%
Liberty	9.64	10.26	-6.04%

Bit correlation reduction. To investigate the reduction of bit correlations brought by ACM, we compare the correlations between bits of binary descriptors derived by the models trained with (w/) and without (w/o) ACM, respectively. Specifically, we train the proposed network on the three subsets of the Brown dataset separately, with the same experimental settings, except that ACM is removed from each model.

With the derived binary descriptors, the average bit correlations are evaluated by *mean Absolute Correlations* (mAC). Specifically, given N to-be-evaluated image patches with corresponding k-bit binary descriptors $B = \{b_1, ...b_n\}$, the mAC score is calculated as follows,

$$mAC = \frac{1}{k(k-1)} \sum_{i,j \neq i} |P_{ij}|, \tag{23}$$

$$P_{ij} = \frac{\sum_{n=1}^{N}(b_{in} - \bar{b_i})(b_{jn} - \bar{b_j})}{\sqrt{\sum_{n=1}^{N}(b_{in} - \bar{b_i})^2}\sqrt{\sum_{n=1}^{N}(b_{jn} - \bar{b_j})^2}}, \tag{24}$$

where P_{ij} presents the *Pearson correlation coefficient* between the i-th bit and the j-th bit. Specifically, b_{in} denotes the i-th bit of the binary descriptor of the n-th image patch. $\bar{b_i}$ and $\bar{b_i}$ are the mean values of the i-th bit and the j-th bit over N image patches. According to the definition of mAC, it can be inferred that *a lower mAC score means lower bit correlations.*

The mAC scores of binary local descriptors derived by models with (w) or without (w/o) ACM are reported in Table VI. As can be seen, with ACM, the correlation between bits is reduced by 3.65%, 22.92%, 6.04%, respectively, in terms of

mAC score, which proves its effectiveness on bits correlation reduction.

V. Conclusion

In this paper, we have proposed a Transformation-invariant Binary Local Descriptor learning method, termed TBLD, which is trained in an unsupervised manner. Three major contributions are made in the paper. First, we proposed a framework that derives transformation-invariant binary local descriptors. Based on the argumentation that the same object should be described by the same local descriptors, the original image patches and their transformed counterparts are projected to an identical Euclidean subspace and an identical Hamming subspace with the help of the contrastive loss. Second, to solve the problem brought by directly applying the scheme of image hashing in local descriptor learning, which refers to the high correlations between bits, we propose an *Adversarial Constraint Module* (ACM). A set of low-coupling binary codes are introduced to guide the learning of binary local descriptors. By means of Wasserstein loss, the framework is optimized to transfer the distribution of the learned binary local descriptors to the low-coupling ones, thereby making the learned ones as low-coupling as possible. Third, experimental results on three benchmark datasets well demonstrate the superiority of the proposed approach over the state-of-the-art methods.

Appendix

A. Image entropy

For the experiment mentioned in Section I, we evaluate the average amount of information conveyed by image patches and images with *Shannon entropy*, which serves as a measurement of image information and is extensively used in image processing applications [45]. The Shannon Entropy is defined as,

$$E = -\sum_{i=1}^{256} p_i log(p_i), \tag{25}$$

where p_i denotes the probability of the i-th gray-level value occurring in an image or an image patch. A higher Shannon entropy means more information are carried.

References

[1] L. Ding, Y. Tian, H. Fan, C. Chen, and T. Huang, "Joint coding of local and global deep features in videos for visual search," *IEEE Transactions on Image Processing*, vol. 29, pp. 3734–3749, 2020.

[2] Y. Shen, R. Ji, K. Yang, C. Deng, and C. Wang, "Category-aware spatial constraint for weakly supervised detection," *IEEE Transactions on Image Processing*, vol. 29, pp. 843–858, 2019.

[3] Y. Zhang, Y. Huang, S. Yu, and L. Wang, "Cross-view gait recognition by discriminative feature learning," *IEEE Transactions on Image Processing*, vol. 29, pp. 1001–1015, 2019.

[4] S. Qiao, R. Wang, S. Shan, and X. Chen, "Deep heterogeneous hashing for face video retrieval," *IEEE Transactions on Image sProcessing*, vol. 29, pp. 1299–1312, 2019.

[5] S. Leutenegger, M. Chli, and R. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *2011 IEEE international conference on computer vision (ICCV)*. Ieee, 2011, pp. 2548–2555.

[6] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *European conference on computer vision*. Springer, 2010, pp. 778–792.

[7] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "Orb: An efficient alternative to sift or surf." in *ICCV*, vol. 11. Citeseer, 2011, p. 2.

[8] T. Trzcinski, M. Christoudias, and V. Lepetit, "Learning image descriptors with boosting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 3, pp. 597–610, 2014.

[9] C. Strecha, A. Bronstein, M. Bronstein, and P. Fua, "Ldahash: Improved matching with smaller descriptors," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 1, pp. 66–78, 2011.

[10] K. Lin, J. Lu, C.-S. Chen, J. Zhou, and M.-T. Sun, "Unsupervised deep learning of compact binary descriptors," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 6, pp. 1501–1514, 2018.

[11] Y. Duan, J. Lu, Z. Wang, J. Feng, and J. Zhou, "Learning deep binary descriptor with multi-quantization." *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1924–1938, 2019.

[12] Y. Tian, B. Fan, and F. Wu, "L2-net: Deep learning of discriminative patch descriptor in euclidean space," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 661–669.

[13] Y. Duan, Z. Wang, J. Lu, X. Lin, and J. Zhou, "Graphbit: Bitwise interaction mining via deep reinforcement learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8270–8279.

[14] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, "Working hard to know your neighbor's margins: Local descriptor learning loss," in *Advances in Neural Information Processing Systems*, 2017, pp. 4826–4837.

[15] J. Ye, S. Zhang, T. Huang, and Y. Rui, "Cdbin: Compact discriminative binary descriptor learned with efficient neural network," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[16] M. Zieba, P. Semberecki, T. El-Gaaly, and T. Trzcinski, "Bingan: Learning compact binary descriptors with a regularized gan," in *Advances in Neural Information Processing Systems*, 2018, pp. 3608–3618.

[17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[18] Y. Li and J. van Gemert, "Deep unsupervised image hashing by maximizing bit entropy," *arXiv preprint arXiv:2012.12334*, 2020.

[19] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[20] M. Brown, G. Hua, and S. Winder, "Discriminative learning of local image descriptors," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 1, pp. 43–57, 2010.

[21] Y. Guo, X. Zhao, G. Ding, and J. Han, "On trivial solution and high correlation problems in deep supervised hashing." AAAI, 2018.

[22] V. Erin Liong, J. Lu, G. Wang, P. Moulin, and J. Zhou, "Deep hashing for compact binary codes learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2475–2483.

[23] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 119. Virtual: PMLR, 13–18 Jul 2020, pp. 1597–1607.

[24] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 12, pp. 2916–2929, 2012.

[25] G. Wu, J. Han, Y. Guo, L. Liu, G. Ding, Q. Ni, and L. Shao, "Unsupervised deep video hashing via balanced code for large-scale video retrieval," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1993–2007, 2018.
[26] L. Liu and L. Shao, "Sequential compact code learning for unsupervised image hashing," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 12, pp. 2526–2536, 2015.
[27] K. He, Y. Lu, and S. Sclaroff, "Local descriptors optimized for average precision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 596–605.
[28] G. Wu, Z. Lin, G. Ding, Q. Ni, and J. Han, "On aggregation of unsupervised deep binary descriptor with weak bits," *IEEE Transactions on Image Processing*, vol. 29, pp. 9266–9278, 2020.
[29] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.
[30] L. Wu, Y. Wang, H. Yin, M. Wang, and L. Shao, "Few-shot deep adversarial learning for video-based person re-identification," *IEEE Transactions on Image Processing*, vol. 29, pp. 1233–1245, 2019.
[31] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Advances in neural information processing systems*, 2016, pp. 2172–2180.
[32] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5907–5915.
[33] Y. Cao, B. Liu, M. Long, and J. Wang, "Hashgan: Deep learning to hash with pair conditional wasserstein gan," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1287–1296.
[34] J. Song, T. He, L. Gao, X. Xu, A. Hanjalic, and H. T. Shen, "Binary generative adversarial networks for image retrieval," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[36] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.
[37] Z. Zheng, X. Yang, Z. Yu, L. Zheng, Y. Yang, and J. Kautz, "Joint discriminative and generative learning for person re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 2138–2147.
[38] M. A. Albahar, A. A. Albahr, and M. H. Binsawad, "An efficient person re-identification model based on new regularization technique," *IEEE Access*, vol. 8, pp. 171 049–171 057, 2020.
[39] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in neural information processing systems*, 2017, pp. 5767–5777.
[40] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning," in *Advances in neural information processing systems*, 2008, pp. 161–168.
[41] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.
[42] T. Trzcinski and V. Lepetit, "Efficient discriminative projections for compact binary descriptors," in *European Conference on Computer Vision*. Springer, 2012, pp. 228–242.
[43] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5173–5182.
[44] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," *International journal of computer vision*, vol. 65, no. 1-2, pp. 43–72, 2005.
[45] Q. Razlighi and N. Kehtarnavaz, "A comparison study of image spatial entropy," in *Visual Communications and Image Processing 2009*, vol. 7257. International Society for Optics and Photonics, 2009, p. 72571X.

Yunqi Miao is currently pursuing the Ph.D. degree with the Warwick Manufacturing Group (WMG), University of Warwick, Coventry, U.K. Her research interests include binary descriptors learning and person re-identification.

Zijia Lin received his Ph.D. degree from Department of Computer Science and Technology, Tsinghua University, Beijing, China in 2016 and his B.Sc. degree from School of Software in the same campus in 2011. His research interests include multimedia information retrieval and machine learning.

Xiao Ma is an Associate Professor at WMG, the University of Warwick. He is also the head of the WMG Accelerator Programme, which has been set up to fast-track growth for innovative technology. His reserach interests include digital economy, technology innovation, entrepreneurship, business and industry transformation.

Guiguang Ding is currently an Associate Professor with the School of Software, Tsinghua University, China. His current research interests include the areas of multimedia information retrieval, computer vision, and machine learning.

Jungong Han is a Chair Professor and the Director of Research of Computer Science at Aberystwyth University, UK. He also holds an Honorary Professorship at the University of Warwick, UK. His research interests include computer vision, artificial intelligence and machine learning.