Generative Domain-Migration Hashing for Sketch-to-Image Retrieval

Anonymous ECCV submission

Paper ID 1917

Abstract. Due to the succinct nature of free-hand sketch drawings, sketch-based image retrieval (SBIR) has abundant practical use cases in consumer electronics. However, SBIR remains a long-standing unsolved problem mainly because of the significant discrepancy between the sketch domain and the image domain. In this work, we propose a Generative Domain-migration Hashing (GDH) approach, which for the first time generates hashing codes from synthetic natural images that are migrated from sketches. The generative model learns a mapping that the distributions of sketches can be indistinguishable from the distribution of natural images using an adversarial loss, and simultaneously learns an inverse mapping based on the cycle consistency loss in order to enhance the indistinguishability. With the robust mapping learned from the generative model, GDH can migrate sketches to their indistinguishable image counterparts while preserving the domain-invariant information of sketches. With an end-to-end multi-task learning framework, the generative model and binarized hashing codes can be jointly optimized. Comprehensive experiments of both category-level and fine-grained SBIR on multiple large-scale datasets demonstrate the consistently balanced superiority of GDH in terms of efficiency, memory costs and effectiveness.

Keywords: Domain-migration · Hash function · SBIR

1 Introduction

The prevalence of touchscreen in consumer electronics (range from portable devices to large home appliance) facilitates human-machine interactions free-hand drawings. The input of sketches is succinct, convenient and efficient for visually recording ideas, and can beat hundreds of words in some scenarios. As an extended application based on sketches, sketch-based image retrieval (SBIR) [1–7] has attracted increasing attention.

The primary challenge in SBIR is that free-hand sketches are inherently abstract and iconic, which magnifies cross-domain discrepancy between sketches and real-world images. Recent works attempt to employ cross-view learning methods [1, 7–15] to address such a challenge, where the common practice is to reduce the domain discrepancy by embedding both sketches and natural images to a common space and use the projected features for retrieval. The most critical deficiency of this line of approaches is the learned mappings within each

domain cannot be well-generalized to the test data, especially for categories with large variance. Similar to other image-based retrieval problems, the query time grows increasingly with the database size and exponentially with the dimension of sketch/image representations. To this end, Deep Sketch Hashing (DSH) [8] is introduced to replace the full-precision sketch/image representations with binary vectors. However, the quantization error introduced by the binarization procedure can destroy both domain-invariant information and the semantic consistency across domains.

In this work, our primary goal is to improve deficiencies in aforementioned works and provide a practical solution to the scalable SBIR problem. We propose a Generative Domain-migration Hashing (GDH) method that improves the generalization capability by migrating sketches into the natural image domain, where the distribution migrated sketches can be indistinguishable from the distribution of natural images. Additionally, we introduce an end-to-end multi-task learning framework that jointly optimizes the cycle consistent migration as well as the hash codes, where the adversarial loss and the cycle consistency loss can simultaneously preserve the semantic consistency of the hashing codes. GDH also integrates an attention layer that guides the learning process to focus on the most representative regions.

While SBIR aims to retrieve natural images that shares identical category labels with the query sketch, fine-grained SBIR aims to preserve the intra-category instance-level consistency in addition to the category-level consistency. For the consistency purpose, we refer to standard SBIR as category-level SBIR and the fine-grained version as fine-grained SBIR respectively throughout the paper. Since the bidirectional mappings learned in GDH are highly under-constrained (*i.e.*, does not require the pixel-level alignment [16] between sketches and natural images), GDH can naturally provide an elegant solution for preserving the geometrical morphology and detailed instance-level characteristic between sketches and natural images. In addition, a triplet ranking loss is introduced to enhance the fine-grained learning based on visual similarities of intra-class instances. The pipeline of the proposed GDH method for both category-level and fine-grained SBIR tasks is illustrated in Fig. 1. Extensive experiments on various large-scale datasets for both category-level and fine-grained SBIR tasks demonstrate the consistently balanced superiority of GDH in terms of memory cost, retrieval time and accuracy. The main contributions of this work are as follows:

- We for the first time propose a generative model GDH for the hashing-based SBIR problem. Comparing to existing methods, the generative model can essentially improve the generalization capability by migrating sketches into their indistinguishable counterparts in the natural image domain.
- Guided by an adversarial loss and a cycle consistency loss, the optimized binary hashing codes can preserve the semantic consistency across domains. Meanwhile, training GDH does not require the pixel-level alignment across domains, and thus allows generalized and practical applications.
- GDH can improve the category-level SBIR performance over the state-of-the-art hashing-based SBIR method DSH [8] by up to 20.5% on the TU-Berlin Extension dataset, and up to 26.4% on the Sketchy dataset respectively.

Meanwhile, GDH can achieve comparable performance with real-valued fine-grained SBIR methods, while significantly reduce the retrieval time and memory cost with binary codes.

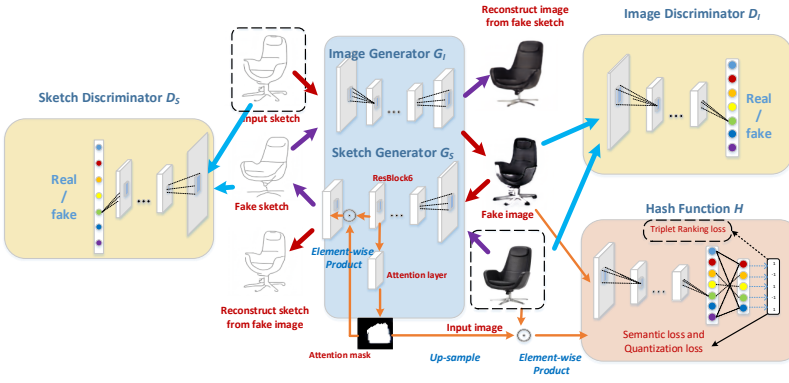

Fig. 1. Illustration of our deep model for the domain-migration networks and compact binary codes learning. The domain-migration module consists of G_I, G_S, D_I and D_S. The bottom right module is the hashing network H. The red arrows represent the cycle between real sketches and fake natural while the purple arrows represent the cycle between real natural images and fake sketches.

2 Related Work

In this section, we discuss the following four directions of related works.

Category-level SBIR: The majority of existing category-level SBIR methods [1, 7–15, 17] rely on learning a common feature space for both sketches and natural images. However, learning such a common feature space based on the can end up with an overfitting solution to the training data.

Hashing-based SBIR: If the learned common feature space is real-valued, the retrieval time depends on the database size, and the scalability of the algorithms can be consequently restrained. In order to to improve the efficiency, hashing-based methods [18–26] are introduced to solve the SBIR problem. The state-of-the-art hashing-based SBIR method DSH [26] employed an end-to-end semi-heterogeneous CNNs to learn binarized hashing codes for retrieval. However, the generalization issue remains in DSH since the learned semi-heterogeneous CNNs are also non-linear mappings across the two domains.

Generative Adversarial Networks: The success of Generative Adversarial Networks (GANs) [27] in various image generation [28] and representation learning [29] tasks is inspiring in a way that sketches can be migrated into the natural image domain using the adversarial loss, where the migrated sketches cannot be distinguished from natural images. Image-to-image translation methods [30, 31] can serve this purpose and are capable of migrating sketches into natural images, however, the pixel-level alignment between each sketch and image pair required for training are impractical. In order to address such an issue, Zhu et al. [32]

introduced a cycle consistency loss. In this work, we employ such a cycle consistency loss and force the bidirectional mappings to be consistent with each other. Benefiting from the highly under-constrained cycled learning, sketches can be migrated to their indistinguishable counterparts in the natural image domain.

Fine-grained SBIR: Among a limited number of fine-grained SBIR methods [4–6, 33–38], Yu et al. [4] proposed the multi-branch networks with triplet ranking loss, which preserved the visual similarities of intra-class sketch and natural image instances. In our work, we also exploit the triplet ranking loss for preserving the visual similarity of intra-class instances. With improved generalization capability to the test data and the binarized hashing codes, the proposed GDH method can achieve comparable performance with [4] on the fine-grained SBIR task, while requiring much less memory and retrieval time.

3 Generative Domain-migration Hash

3.1 Preliminary

Given n_1 training images $I = \{I_i\}_{i=1}^{n_1}$ and n_2 training sketches $S = \{S_i\}_{i=1}^{n_2}$, the label vectors (row vectors) for all the training instances are $Y^I = \{\mathbf{y}_i^I\}_{i=1}^{n_1} \in \{0,1\}^{n_1 \times c}$ and $Y^S = \{\mathbf{y}_i^S\}_{i=1}^{n_2} \in \{0,1\}^{n_2 \times c}$, respectively, where $\mathbf{y}_i^I$ and $\mathbf{y}_i^S$ are one-hot vectors and c is the number of classes. We aim to learn the migration from sketches to natural images, and simultaneously learn the hashing function $\mathbf{H} : \{I, I_{fake}\} \rightarrow \{-1, +1\}^K$, such that the semantic consistency can be preserved between the extracted hashing codes of both authentic and generated natural images.

3.2 Network Architecture

To serve the above purposes, we simultaneously optimize a pair of generative and discriminative networks and a hashing network.

Generative networks: Let G_I and G_S be two parallel generative CNNs for migrating sketches to the natural images and vice versa: $G_I : S \rightarrow I$ and $G_S : I \rightarrow S$. The detailed architectures of CNNs are illustrated in Table 1. Considering natural images contain much more information than their sketch counterparts, migrating sketches to natural images is essentially an upsampling process and potentially requires more parameters.

In order to suppress the background information and guide the learning process to concentrate on the most representative regions, we integrate an attention module [39, 40] in G_S. The attention module contains a convolutional layer with 1×1 kernel size, where a softmax function with a threshold is applied to the output for obtaining a binary attention mask. Element-wise $\odot$ multiplication can be performed between the binary attention mask and the feature map from ResBlocks.

Discriminative networks: Along with two generators, two discriminative networks are correspondingly integrated in GDH, where D_I aims to distinguish the

images with its mask ($I \odot mask$) and the generated images $G_I(S)$, and D_S aims to distinguish the real sketches S and the generated sketches $G_S(I)$.

Hashing network: The hashing network $\mathbf{H}$ aims to generate binary hashing codes of both real images I and generated images $G_I(S)$, and can be trained based on both real image with its mask ($I \odot mask$) and generated image $G_I(S)$ from the domain-migration network. The hashing network $\mathbf{H}$ is modified from the 18-layer Deep Residual Network (Resnet) [41] by replacing the softmax layer with a fully-connected layer with a binary constraint on the values, where the dimension of the fully-connected layer equals to the length of the hashing codes.

We denote the parameters of the shared-weight hashing network as $\boldsymbol{\theta}_H$. For natural images and sketches, we formulate the deep hash function (*i.e., the Hashing network*) as $\mathbf{B}^I = \text{sgn}(\mathbf{H}(I \odot mask; \boldsymbol{\theta}_H)) \in \{0,1\}^{n_1 \times K}$ and $\mathbf{B}^S = \text{sgn}(\mathbf{H}(G_I(S); \boldsymbol{\theta}_H)) \in \{0,1\}^{n_2 \times K}$, respectively, where $\text{sgn}(\cdot)$ is the sign function. Note that we use the row vector of the output for the convenience of computation. In the following section, we will introduce the deep generative hashing objective of joint learning of binary codes and hash functions.

Table 1. Network architecture of the generative domain-migration network.

Net	Layer	Kernel Size	Stride	Pad	Output	Net	Layer	Kernel Size	Stride	Pad	Output
G_S-Net	input	-	-	-	3×256×256	G_I-Net	input	-	-	-	3×256×256
	conv1 BN	3×3	1	1	32×256×256		*conv1 BN*	3×3	1	1	32×256×256
	conv2 BN	3×3	1	1	32×256×256		*conv1 BN*	3×3	1	1	32×256×256
	conv3 BN	4×4	2	1	64×128×128		*conv2 BN*	4×4	2	1	64×128×128
	conv4 BN	4×4	2	1	128×64×64		*conv3 BN*	4×4	2	1	128×64×64
	**ResBlock*×6*	3×3	1	1	128×64×64		**ResBlock*×9*	3×3	1	1	128×64×64
	attention	1×1	1	0	1×64×64		*Deconv4 BN*	4×4	2	1	64×128×128
	Deconv4 BN	4×4	2	1	64×128×128		*Deconv5 BN*	4×4	1	1	32×253×253
	Deconv5 BN	4×4	2	1	32×256×256		*Deconv6 BN*	3×3	1	1	3×256×256
	Deconv6 BN	3×3	1	1	3×256×256						
D_S-Net	input	-	-	-	3×256×256	D_I-Net	input	-	-	-	3×256×256
	conv1 BN	4×4	2	1	64×128×128		*conv1 BN*	4×4	2	1	64×128×128
	conv2 BN	4×4	2	1	128×64×64		*conv2 BN*	4×4	2	1	128×64×64
	conv3 BN	4×4	2	1	256×32×32		*conv3 BN*	4×4	2	1	256×32×32
	conv4 BN	3×3	1	1	512×32×32		*conv4 BN*	3×3	1	1	512×32×32
	conv5 BN	3×3	1	1	128×32×32		*conv5 BN*	3×3	1	1	128×32×32

The ResBlock is constructed out of two *conv* layers with *BN* and a pass-through through the information from previous layers unchanged. The hyper-parameters of the *conv* layer are: Kernel Size 3×3, Stride 1, Padding 1 and Channels 128.

3.3 Objective Formulation

There are five losses in our objective function. The adversarial loss and the cycle consistency loss guide the learning of the domain-migration network. The semantic and triplet losses preserve the semantic consistency and visual similarity of intra-class instances across domains. The quantization loss and unification constraint can preserve the feature space similarity of pair instances. Detailed discussion of each loss is provided in following paragraphs.

Adversarial and Cycle Consistency Loss: Our domain-migration networks are composed of four parts: G_I, G_S, D_I and D_S [32]. We denote the parameters of G_I, G_S, D_I and D_S as $\boldsymbol{\theta}_C$. Specifically, $\boldsymbol{\theta}_C|_{G_I}$ is the parameter of G_I and so forth. Note that the inputs of domain-migration networks should be image-sketch pairs

and usually we have $n_1 \gg n_2$. Thus we reuse the sketches from same category to match the images. Sketches from the same category are randomly repeated and S will be expanded to $\hat{S} = \{S_1, \cdots, S_1, S_2 \cdots, S_2, \cdots, S_{n_2}, \cdots, S_{n_2}\}$ to make sure $|\hat{S}| = |I|$. Suppose the data distributions are $I \sim p_I$ and $\hat{S} \sim p_{\hat{S}}$. For the generator $G_I : \hat{S} \rightarrow I$ and its discriminator D_I, the adversarial loss can be written as

$$\begin{aligned} \min_{\boldsymbol{\theta}_C|_{G_I}} \max_{\boldsymbol{\theta}_C|_{D_I}} \mathcal{L}_G(G_I, D_I, \hat{S}, I) := & \mathbf{E}_{I \sim p_I}[\log D_I(I \odot mask, \boldsymbol{\theta}_C|_{D_I})] \\ & + \mathbf{E}_{\hat{S} \sim p_{\hat{S}}}[\log(1 - D_I(G_I(\hat{S}, \boldsymbol{\theta}_C|_{G_I}), \boldsymbol{\theta}_C|_{D_I})], \end{aligned} \tag{1}$$

where the generator and the discriminator compete in a two-player minimax game: the generator tries to generate images $G_I(\hat{S})$ that look similar to the images from domain I and its corresponding $mask$, while the discriminator tries to distinguish between real images and fake images. The adversarial loss of the other mapping function $G_S : I \rightarrow \hat{S}$ is defined in the similar way. The Cycle Consistency Loss can prevent the learned mapping function G_I and G_S from conflicting against each other, which can be expressed as

$$\begin{aligned} \min_{\boldsymbol{\theta}_C|_{G_I}, \boldsymbol{\theta}_C|_{G_S}} \mathcal{L}_{cyc}(G_I, G_S) := & \mathbf{E}_{I \sim p_I} \| G_S(G_I(\hat{S}, \boldsymbol{\theta}_C|_{G_I}), \boldsymbol{\theta}_C|_{G_S}) - \hat{S} \| \\ & + \mathbf{E}_{\hat{S} \sim p_{\hat{S}}} \| G_I(G_S(I, \boldsymbol{\theta}_C|_{G_S}), \boldsymbol{\theta}_C|_{G_I}) - I \odot mask \|. \end{aligned} \tag{2}$$

where $\| \cdot \|$ is the Frobenius norm. The full optimization problem for domain-migration networks is

$$\min_{\substack{\boldsymbol{\theta}_C|_{G_I} \\ \boldsymbol{\theta}_C|_{G_S}}} \max_{\substack{\boldsymbol{\theta}_C|_{D_I} \\ \boldsymbol{\theta}_C|_{D_S}}} \mathcal{L}_{gan} := \mathcal{L}_G(G_I, D_I, \hat{S}, I) + \mathcal{L}_G(G_S, D_S, I, \hat{S}) + \upsilon \mathcal{L}_{cyc}(G_I, G_S). \tag{3}$$

We set the balance parameter $\upsilon = 10$ in the experiment according to the previous work [32].

Semantic Loss: The label vectors of images and sketches are Y^I and Y^S. Inspired by Fast Supervised Discrete Hashing [42], we consider the following semantic factorization problem with the projection matrix $\mathbf{D} \in \mathbb{R}^{c \times K}$:

$$\begin{aligned} \min_{\mathbf{B}^I, \mathbf{B}^S, \mathbf{D}} \mathcal{L}_{sem} := & \left\| \mathbf{B}^I - Y^I \mathbf{D} \right\|^2 + \left\| \mathbf{B}^S - Y^S \mathbf{D} \right\|^2 + \| \mathbf{D} \|^2, \\ \text{s.t.} \quad & \mathbf{B}^I \in \{-1, +1\}^{n_1 \times K}, \mathbf{B}^S \in \{-1, +1\}^{n_2 \times K}. \end{aligned} \tag{4}$$

$\mathcal{L}_{sem}$ aims to minimize the distance between the binary codes of the same category, and maximize the distance between the binary codes of different categories.

Quantization Loss: The quantization loss is introduced to preserve the intrinsic structure of the data, and can be formulated as follows:

$$\min_{\boldsymbol{\theta}_H} \mathcal{L}_q := \left\| \mathbf{H}\left(I; \boldsymbol{\theta}_H\right) - \mathbf{B}^I \right\|^2 + \left\| \mathbf{H}\left(G_I\left(S, \boldsymbol{\theta}_C|_{G_I}\right); \boldsymbol{\theta}_H\right) - \mathbf{B}^S \right\|^2. \tag{5}$$

Triplet Ranking Loss: For the fine-grained retrieval task, we integrate the triplet ranking loss into the objective function for preserving the similarity of

paired cross-domain instances within an object category. For a given triplet $\left(S_i, I_i^+, I_i^-\right)$, specifically, each triplet contains a query sketch S_i and a positive image sample I_i^+ and a negative image sample I_i^-. We define the triplet ranking loss function as follow:

$$\begin{aligned}\min_{\boldsymbol{\theta}_H} \mathcal{L}_{tri} := \sum_i \max\big(0, \Delta + \left\|\mathbf{H}\left(G_I\left(S_i, \boldsymbol{\theta}_C|_{G_I}\right); \boldsymbol{\theta}_H\right) - \mathbf{H}\left(I_i^+; \boldsymbol{\theta}_H\right)\right\|^2 \\ - \left\|\mathbf{H}\left(G_I\left(S_i, \boldsymbol{\theta}_C|_{G_I}\right); \boldsymbol{\theta}_H\right) - \mathbf{H}\left(I_i^-; \boldsymbol{\theta}_H\right)\right\|^2\big),\end{aligned} \tag{6}$$

where the parameter Δ represents the margin between the similarities of the outputs of the two pairs (S_i, I_i^+) and (S_i, I_i^-). In other words, the hashing network ensures that the Hamming distance between the outputs of the negative pair (S_i, I_i^-) is larger than the Hamming distance between the outputs of the positive pair (S_i, I_i^+) by at least a margin of Δ. In this paper, we let Δ equal to half of the code length (*i.e.*, $\Delta = 0.5K$).

Full Objective Function: We also desire the binary codes of a real natural image and a generated image to be close to each other. Thus, we employ a unification constraint $\mathcal{L}_c = \|\mathbf{H}(I; \boldsymbol{\theta}_H) - \mathbf{H}(G_I(\hat{S}, \boldsymbol{\theta}_C|_{G_I}); \boldsymbol{\theta}_H)\|^2$ is added to the final objective function which is formulated as follows:

$$\begin{aligned}\min_{\mathbf{B^I}, \mathbf{B^S}, \mathbf{D}, \boldsymbol{\theta}_C, \boldsymbol{\theta}_H} \mathcal{L}_{total} := \mathcal{L}_{gan} + \mathcal{L}_{sem} + \lambda \mathcal{L}_{tri} + \alpha \mathcal{L}_q + \beta \mathcal{L}_c, \\ \text{s.t. } \mathbf{B}^I \in \{-1, +1\}^{n_1 \times K}, \mathbf{B}^S \in \{-1, +1\}^{n_2 \times K},\end{aligned} \tag{7}$$

where λ is a control parameter, which equals 1 for fine-grained task and equals 0 for semantic-level SBIR only, The hyper-parameters α and β control the contributions of the two corresponding terms.

3.4 Joint Optimization

Due to the non-convexity of the joint optimization and **NP**-hardness to output the discrete binary codes, it is infeasible to find the global optimal solution. Inspired by [42], we propose an optimization algorithm based on alternating iteration and sequentially optimize one variable while the others are fixed. In this way, variables $\mathbf{D}$, $\mathbf{B}^I$, $\mathbf{B}^S$, parameter $\boldsymbol{\theta}_C$ of the domain-migration networks, and parameter $\boldsymbol{\theta}_H$ of the hash function will be iteratively updated.

$\mathbf{D}$-*Step.* By fixing all the variables except $\mathbf{D}$, Eq. (7) can be simplified as a classic quadratic regression problem:

$$\begin{aligned}&\min_{\mathbf{D}} \left\|\mathbf{B}^I - Y^I\mathbf{D}\right\|^2 + \left\|\mathbf{B}^S - Y^S\mathbf{D}\right\|^2 + \|\mathbf{D}\|^2 \\ &= \min_{\mathbf{D}} tr\left(\mathbf{D}^\top\left({Y^I}^\top Y^I + {Y^S}^\top Y^S + \mathbf{I}\right)\mathbf{D}\right) - 2tr\left(\mathbf{D}^\top\left({Y^I}^\top \mathbf{B}^I + {Y^S}^\top \mathbf{B}^S\right)\right),\end{aligned} \tag{8}$$

where $\mathbf{I}$ is the identity matrix. Taking the derivative of the above function with respect to $\mathbf{D}$ and setting it to zero, we have the analytical solution to Eq. (8):

$$\mathbf{D} = \left({Y^I}^\top Y^I + {Y^S}^\top Y^S + \mathbf{I}\right)^{-1}\left({Y^I}^\top \mathbf{B}^I + {Y^S}^\top \mathbf{B}^S\right). \tag{9}$$

$\mathbf{B}^I$-*Step.* When all the variables are fixed except $\mathbf{B}^I$, we rewrite Eq. (7) as

$$\min_{\mathbf{B}^I} \left\|\mathbf{B}^I - Y^I\mathbf{D}\right\|^2 + \alpha \left\|\mathbf{H}\left(I;\boldsymbol{\theta}_H\right) - \mathbf{B}^I\right\|^2. \tag{10}$$

Since $tr\left(\mathbf{B}^{I\top}\mathbf{B}^I\right)$ is a constant, Eq. (10) is equivalent to the following problem:

$$\min_{\mathbf{B}^I} -tr\left(\mathbf{B}^{I\top}\left(Y^I\mathbf{D} + \alpha\mathbf{H}\left(I;\boldsymbol{\theta}_H\right)\right)\right). \tag{11}$$

For $\mathbf{B}^I \in \{-1,+1\}^{n_1\times K}$, $\mathbf{B}^I$ has a closed-form solution as follows:

$$\mathbf{B}^I = \mathrm{sgn}\left(Y^I\mathbf{D} + \alpha\mathbf{H}\left(I;\boldsymbol{\theta}_H\right)\right). \tag{12}$$

$\mathbf{B}^S$-*Step.* Considering all the terms related to $\mathbf{B}^S$, it can be learned by a similar formulation as Eq.(12):

$$\mathbf{B}^S = \mathrm{sgn}\left(Y^S\mathbf{D} + \alpha\mathbf{H}\left(G_I\left(S,\boldsymbol{\theta}_C|_{G_I}\right);\boldsymbol{\theta}_H\right)\right). \tag{13}$$

$(\boldsymbol{\theta}_C,\boldsymbol{\theta}_H)$-*Step.* After the optimization for $\mathbf{D}$, $\mathbf{B}^I$ and $\mathbf{B}^S$, we update the network parameters $\boldsymbol{\theta}_C$ and $\boldsymbol{\theta}_H$ according to the following loss:

$$\min_{\boldsymbol{\theta}_C,\boldsymbol{\theta}_H} \mathcal{L} := \mathcal{L}_{gan} + \lambda\mathcal{L}_{tri} + \alpha\mathcal{L}_q + \beta\mathcal{L}_c. \tag{14}$$

We train our networks on I and $\hat{S}$, where the sketch-image pairs are randomly select to compose of the mini-batch, and then backpropagation algorithm with SGD is adopted for optimizing two networks. In practice, we use deep learning frameworks (*e.g.*, Pytorch) to achieve all the steps. We iteratively update $\mathbf{D} \rightarrow \mathbf{B}^I \rightarrow \mathbf{B}^S \rightarrow \{\boldsymbol{\theta}_C,\boldsymbol{\theta}_H\}$ in each epoch. As such, GDH can be finally optimized within L epochs, where $20 \leq L \leq 30$ in our experiment. The algorithm of GDH is illustrated in Algorithm 1.

Once GDH model is learned, for a given query sketch s_q, we can infer its binary code $\mathbf{b}^{s_q} = \mathrm{sgn}\left(\mathbf{H}\left(G_I\left(S_q,\boldsymbol{\theta}_C|_{G_I}\right);\boldsymbol{\theta}_H\right)\right)$ through the G_I network and the hash network $\mathbf{H}$. For the image gallery, the hash codes $\mathbf{b}^I = \mathrm{sgn}\left(\mathbf{H}\left(I \odot mask;\boldsymbol{\theta}_H\right)\right)$ of each image is computed through the hash network, where *mask* can be easily obtained by $G_S\left(I;\boldsymbol{\theta}_C|_{G_S}\right)$. Note that fake images generated by $G_I\left(S_q,\boldsymbol{\theta}_C|_{G_I}\right)$ are non-background and thus they don't need multiply *mask* before feed into the hashing network.

4 Experiments and Results

In the experiment section, we aim to address the following three questions:

- How does GDH perform as compared to other state-of-the-art binary or real-valued methods for category-level SBIR?
- How does GDH perform as compared to other state-of-the-art real-valued methods for fine-grained SBIR?
- How does each component or constraint contribute to the overall performance of GDH?

Algorithm 1: Generative Domain-migration Hash (GDH)

Input: Training natural images $I = \{I_i\}_{i=1}^{n_1}$ and the corresponding sketches $S = \{S_i\}_{i=1}^{n_2}$; the label information Y^I and Y^S; the code length K; the number of training epochs L; the balance parameters α, β, λ.

Output: Generative models G_I and G_S; deep hash function $\mathbf{H}$.

Randomly initialize $\mathbf{B}^I \in \{-1, +1\}^{n_1 \times K}$ and $\mathbf{B}^S \in \{-1, +1\}^{n_2 \times K}$;

For $l = 1, 2, \cdots, L$ **do**

  Update $\mathbf{D}$ according to Eq. (9);

  Update $\mathbf{B}^I$ according to Eq. (12);

  Update $\mathbf{B}^S$ according to Eq. (13);

  Update the network parameters $\boldsymbol{\theta}_C$ and $\boldsymbol{\theta}_H$ according to Eq. (14) by training with the l-th epoch data;

End

Return the network parameters $\boldsymbol{\theta}_C$ and $\boldsymbol{\theta}_H$.

4.1 Datasets and Settings

Category-level Retrieval. GDH is evaluated on two largest SBIR datasets: Sketchy [5] and TU-Berlin [43] Extension. The Sketchy database contains 125 categories with 75,471 sketches of 12,500 object images. We additionally utilize another 60,502 natural images [8] collected from ImageNet [44]. Hence, the whole image database contains 73,002 images in total. TU-Berlin is a sketch dataset with 250 object categories, where each category contains 80 sketches. An additional 204,489 natural images associated with TU-Berlin provided by [45] are used to construct the image database. Similar to previous hashing experiments [8], 50 and 10 sketches are respectively selected as the query sets for TU-Berlin and Sketchy, where the remaining are used as the gallery for training.

We compare GDH with 8 existing category-level SBIR methods, including 4 hand-crafted methods: LSK [9], SEHLO [14], GF-HOG [10] and HOG [12]; and 4 deep learning based methods: 3D shape [46], Sketch-a-Net (SaN) [2], GN Triplet [5] and Siamese CNN [36]. Furthermore, we also compare GDH with 7 state-of-the-art cross-modality hashing methods: Collective Matrix Factorization Hashing (CMFH) [47], Cross-Model Semi-Supervised Hashing (CMSSH) [48], Cross-View Hashing(CVH) [49], Semantic Correlation Maximization (SCMSeq and SCM-Orth) [50], Semantics-Preserving Hashing(SePH) [51], Deep Cross-Modality Hashing (DCMH) [52] and Deep Sketch Hash (DSH) [8]. Finally, we also compare our method to other four cross-view feature embedding methods: CCA [53], PLSR [54], XQDA [55] and CVFL [56]. The implementation details and experimental results of above methods are reported in [8].

We use the Adam solver [57] with a batch size of 32. Our balance parameters are set to $\alpha = 10^{-5}, \beta = 10^{-5}$ and $\lambda = 0$ for both datasets. All networks are trained with an initial learning rate $lr = 0.0002$. After 25 epochs, we decrease the learning rate of the hashing network $lr \rightarrow 0.1lr$ and terminate the optimization after 30 epochs for both datasets. Our method is implemented by Pytorch with dual 1080Ti GPUs and an i7-4790K CPU.

Table 2. Comparison with previous SBIR methods w.r.t. MAP, retrieval time per query (s) and memory cost (MB) on on TU-Berlin Extension and Sketchy.

Methods	Dimension	TU-Berlin Extension			Sketchy		
		MAP	Retrieval time per query (s)	Memory cost (MB) (204,489 images)	MAP	Retrieval time per query (s)	Memory cost (MB) (73,002 images)
HOG [12]	1296	0.091	1.43	2.02×10^3	0.115	0.53	7.22×10^2
GF-HOG [10]	3500	0.119	4.13	5.46×10^3	0.157	1.41	1.95×10^3
SHELO [14]	1296	0.123	1.44	2.02×10^3	0.182	0.50	7.22×10^2
LKS [9]	1350	0.157	0.204	2.11×10^3	0.190	0.56	7.52×10^2
Siamese CNN [36]	64	0.322	7.70×10^{-2}	99.8	0.481	2.76×10^{-2}	35.4
SaN [2]	512	0.154	0.53	7.98×10^2	0.208	0.21	2.85×10^2
GN Triplet* [5]	1024	0.187	1.02	1.60×10^3	0.529	0.41	5.70×10^2
3D shape* [46]	64	0.072	7.53×10^{-2}	99.8	0.084	2.64×10^{-2}	35.6
Siamese-AlexNet	4096	0.367	5.35	6.39×10^3	0.518	1.68	2.28×10^3
Triplet-AlexNet	4096	0.448	5.35	6.39×10^3	0.573	1.68 s	2.28×10^3
GDH (Proposed)	32 (bits)	0.563	5.57×10^{-4}	0.78	0.724	2.55×10^{-4}	0.28
	64 (bits)	**0.690**	7.03×10^{-4}	1.56	**0.810**	2.82×10^{-4}	0.56
	128 (bits)	0.659	1.05×10^{-3}	3.12	0.784	3.53×10^{-4}	1.11

"$*$" denotes that we directly use the public models provided by the original papers without any fine-tuning on the TU-Berlin Extension and Sketchy datasets.

Fine-grained Retrieval. We conduct experiments of GDH on the QMUL-Shoes and QMUL-Chairs datasets [4]. The two datasets are fine-grained instance-level SBIR datasets which contain 419 shoes sketch-photo pairs and 297 chairs sketch-photo pairs, respectively.

We compare our proposed GDH method with several fine-grained methods including 2 hand-crafted methods: HOG+BoW+RankSVM [58] and Dense HOG+RankSVM [4], and 3 deep feature baselines: Improved Sketch-a-Net (ISN) [2], 3D shape (3DS) [46] and Triplet Sketch-a-Net (TSN) [4]. All of these algorithms are real-valued methods. It is noteworthy that the networks in TSN [4] are heavily pre-trained and the data have been processed by complex augmentation. However, to emphasize the ability of our domain-migration model, data augmentation is not included in our experiment.

Note that, QMUL-Shoes and QMUL-Chairs are unique fine-grained datasets, in which only contains one category for each of them. Therefore, it is unnecessary to optimize the semantic loss in Eq. (7). To better fit the task of fine-grained retrieval, we skip the first five steps in Algorithm 1 and directly update the parameters of $\boldsymbol{\theta}_C$ and $\boldsymbol{\theta}_H$. Our balance parameters are set to $\lambda = 1$. The implementation details are the same as the settings for category-level retrieval, except that the batch size is 64 and the optimization will be terminated after 200 epochs.

4.2 Results and Discussions

Comparison with Category-level SBIR Baselines. We compare our GDH method with the 10 baseline methods in terms of Mean Average Precision (MAP), retrieval time and memory cost on two datasets. The code lengths of outputs are 32, 64 and 128 bits. As reported in Table 2, GDH consistently achieves the best performance with much faster query time and much lower memory cost compared to other SBIR methods on both datasets. Also, GDH largely improves the state-of-the-art performance of Triplet-AlexNet by 24.2% and 23.7% on the TU-Berlin and Sketchy datasets, respectively. The performance of 128 bits is

Table 3. MAP comparison with different cross-modality retrieval methods for category-level SBIR on TU-Berlin Extension and Sketchy.

Method		TU-Berlin Extension			Sketchy		
		32 bits	64 bits	128 bits	32 bits	64 bits	128 bits
Cross-Modality Hashing Methods (binary codes)	CMFH [47]	0.149	0.202	0.180	0.320	0.490	0.190
	CMSSH [48]	0.121	0.183	0.175	0.206	0.211	0.211
	SCM-Seq [50]	0.211	0.276	0.332	0.306	0.417	0.671
	SCM-Orth [50]	0.217	0.301	0.263	0.346	0.536	0.616
	CVH [49]	0.214	0.294	0.318	0.325	0.525	0.624
	SePH [51]	0.198	0.270	0.282	0.534	0.607	0.640
	DCMH [52]	0.274	0.382	0.425	0.560	0.622	0.656
	DSH [8]	0.358	0.521	0.570	0.653	0.711	0.783
Cross-View Feature Learning Methods (real-valued vectors)	CCA [53]	0.276	0.366	0.365	0.361	0.555	0.705
	XQDA [54]	0.191	0.197	0.201	0.460	0.557	0.550
	PLSR [55]	0.141 (4096-d)			0.462 (4096-d)		
	CVFL [56]	0.289 (4096-d)			0.675 (4096-d)		
Proposed	**GDH**	**0.563**	**0.690**	**0.651**	**0.724**	**0.811**	**0.784**

For end-to-end deep methods, raw natural images and sketches are used. For others, 4096-d AlexNet *fc7* image features and 512-d SaN *fc7* sketch features are used. PLSR and CVFL are both based on reconstructing partial data to approximate full data, so the dimensions are fixed to 4096-d.

lower than the performance of 64 bits can be explained with the quantization error accumulation [21]. We also notice that the performance of compared methods on both datasets is much lower than reported in previous papers[46, 4]. The reason is that the data they previously used are all well-aligned with perfect background removal and the edge of objects can almost fit the sketches. Meanwhile, our experiments adopt realistic images with complicated background, which are greatly different from sketches.

Comparison with Cross-modality Hashing. In Table 3, we compare our GDH method with cross-modality hashing/feature learning methods with 32, 64 and 128 bits binary codes. We use the learned deep features as the inputs for non-end-to-end learning methods for a fair comparison with GDH. GDH achieves the best performance compared to all the cross-modality baselines on both datasets. Specifically, GDH can outperform the best-performing hashing-based SBIR method DSH [8]by 20.5%/7.1%, 16.9%/10% and 8.1%/0.1% at different code lengths on both datasets, respectively. In addition, we illustrate the t-SNE visualization in Fig. 2 on the Sketchy dataset, which shows the hashing codes of query sketches and the image gallery share common distributions in the embedded space.

Comparison for Fine-grained SBIR. In Table 4, we report the top-1 and top-10 accuracies of GDH over other five methods on the Shoes and Chairs datasets for fine-grained SBIR. Compared to the state-of-the-art real-valued TSN (without data augmentation), the 128-bit GDH achieves 2.7%/2.7% and 2.6%/3.4% improvements in terms of top-1 and top-10 accuracies on both the Shoes and Chairs datasets respectively. Specifically, the top-10 accuracy on the Chairs dataset reaches 99%, which is even higher than the performance of TSN with data augmentation.

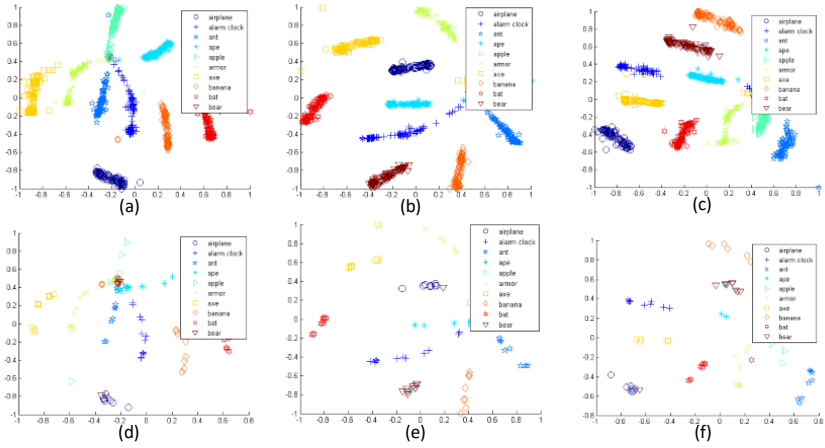

Fig. 2. t-SNE visualization of GDH for 10 representative categories in the Sketchy dataset: (a-c) binary codes of images at 32, 64 and 128 bits; (d-f) binary codes of sketches at 32, 64 and 128 bits. The embedded hash codes of both image and sketch domains share the same distribution and discriminative ability.

Table 4. Accuracy comparison with different real-valued methods for fine-grained SBIR on QMUL-shoes and QMUL-chairs.

	Methods	QMUL-shoes.acc@1	QMUL-shoes.acc@10	QMUL-chairs.@1	QMUL-chairs.@10
Real-valued vectors	BoW-HOG + rankSVM [58]	0.174	0.678	0.289	0.670
	Dense-HOG + rankSVM [4]	0.244	0.652	0.526	0.938
	ISN Deep + rankSVM [2]	0.200	0.626	0.474	0.825
	3DS Deep + rankSVM [46]	0.052	0.217	0.061	0.268
	TSN without data aug. [4]	0.330	0.817	0.644	0.956
	TSN with data aug. [4]	0.391	0.878	0.691	0.979
Binary codes	GDH @ 32-bit	0.286	0.720	0.392	0.876
	GDH @ 64-bit	0.323	0.783	0.556	0.959
	GDH @ 128-bit	0.357	0.843	0.671	0.990

To emphasize the ability of our domain-migration model, data augmentation [4] is not included. Even so, our binary results are competitive and promising compared to other real-valued methods.

Remark. For fine-grained SBIR, despite binary hashing codes are used, comparable or even improved performance over the real-valued state-of-the-art methods can be observed in Table. 4. On the other side, the binary codes in GDH allow much reduced memory costs and retrieval time than the real-valued approaches. However, GDH generally shows degraded performance on the fine-grained SBIR when comparing to its performance on category-level SBIR. Our explanation towards such a phenomenon is that the geometrical morphology and detailed instance-level characteristic within a category can be much more difficult to capture with binary hashing codes than the inter-category discrepancies. In Fig. 3, some examples based on the retrieval results of GDH are illustrated. More illustrations can be found in the **Supplementary Material**.

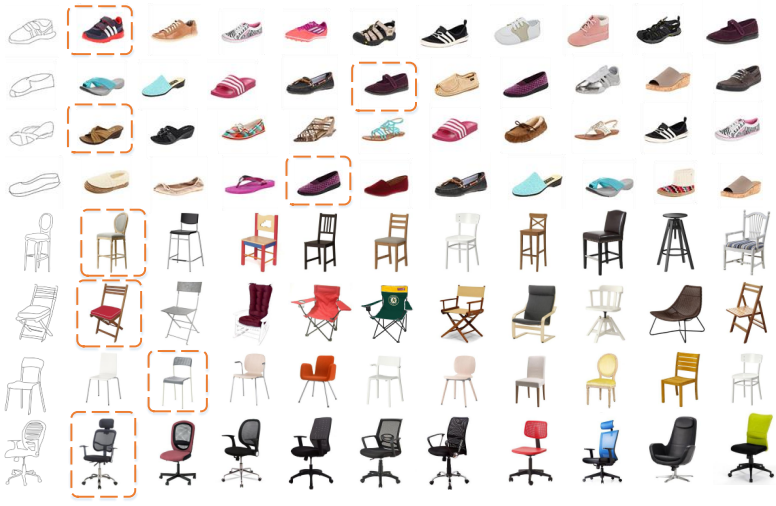

Fig. 3. Example query sketches with their top-10 retrieval accuracies on the Sketchy dataset by using 128-bit GDH codes. Orange boxes indicate the groundtruth results.

4.3 Ablation Study

We demonstrate the effectiveness of each loss component of GDH in Table 5. The detailed descriptions of the unification constraint $\mathcal{L}_c$, the quantization loss $\mathcal{L}_q$ and the adversarial and cycle consistent loss $\mathcal{L}_{gan}$ are provided in Section 3.3. It can be observed that all these components are complementary and beneficial to the effectiveness of GDH. Especially, the adversarial and cycle consistent loss $\mathcal{L}_{gan}$ and the quantization loss $\mathcal{L}_q$ are equivalently critical for category-level SBIR, and he triplet ranking loss $\mathcal{L}_{tri}$ is essential for fine-grained SBIR. It can also be observed that the attention layer is consistently effective for improving the overall performance with a stable margin.

Inspired by the mix-up operation [59], in order to further reduce the domain discrepancy, we propose a feature fusion method that employs a linear mix-up of two types of hashing binary codes: 1) $\text{sgn}(\frac{1}{2}\mathbf{H}(G_I(G_S(I_i, \boldsymbol{\theta}_C|_{G_I}), \boldsymbol{\theta}_C|_{G_S}); \boldsymbol{\theta}_H) + \frac{1}{2}\mathbf{H}(I_i; \boldsymbol{\theta}_H))$ and 2) $\text{sgn}(\mathbf{H}(G_I(S_i, \boldsymbol{\theta}_C|_{G_I}); \boldsymbol{\theta}_H))$. Besides the linear embedding, we also evaluated other fusion strategies such as concatenation and the Kronecker product. However, none of these fusion methods is helpful. In Fig. 4, we illustrate that the generated sketches of GDH can well represent corresponding natural images. It is obviously observed that using sketches to generate fake natural images are more difficult than the inverse generation. Additionally, we conduct another experiment in the sketch domain rather than the natural image domain. By using a similar hashing technique in the sketch domain, all the sketches S and the corresponding generated fake sketches $G_S(I)$ are embedded into the Hamming space as $\mathbf{H}(S_i) = \mathbf{H}(S_i; \boldsymbol{\theta}_H)$ and $\mathbf{H}(I_i) = \mathbf{H}(G_S(I_i); \boldsymbol{\theta}_H)$. However, it resulted in a dramatically decreased performance, especially when handling images that have complex backgrounds.

Fig. 4. Visualization of our domain-migration networks. The first two rows are sketch-to-image results and the last two rows are image-to-sketch results, which indicates that our domain-migration networks are capable to transfer domains from both directions.

Table 5. Effectiveness (MAP/accuracy with 128-bit) of different components (Sketchy for category-level SBIR and QMUL-Shoes for fine-grained SBIR).

Methods	Category-level MAP (Sketchy)	Fine-grained acc. (QMUL-Shoes) top-1	top-10
without $\mathcal{L}_c$	0.727	-	-
without $\mathcal{L}_q$	0.104	-	-
without $\mathcal{L}_{gan}$	0.221	0.226	0.671
without attention layer	0.798	0.335	0.823
Linear mix-up	0.782	0.282	0.744
Concatenation mix-up	0.642	0.182	0.654
Kronecker product mix-up	0.735	0.242	0.704
Embed images into sketch domain	0.310	0.263	0.791
Our model GDH @ 128-bit (binary)	**0.811**	**0.357**	**0.843**

5 Conclusion

In this paper, we proposed a Generative Domain-migration Hashing method for both category-level and fine-grained SBIR tasks. Instead of mapping sketches and natural images into a common space, GDH for the first time employs a generative model that migrates sketches to their indistinguishable counterparts in the natural image domain. Guided by the adversarial loss and the cycle consistency loss, robust hashing codes for both real and synthetic images (*i.e., migrated from sketches*) can be obtained with an end-to-end multi-task learning framework that does not rely on the pixel-level alignment between cross-domain pairs. We additionally integrated an attention layer to effectively suppress the background information and guide the learning process of GDH to concentrate on the most critical regions. Extensive experiments on large-scale datasets demonstrated the consistently balanced superiority of GDH in terms of efficiency, memory costs and performance on both category-level and fine-grained SBIR tasks. GDH also outperformed the best-performing hashing-based SBIR method DSH [8] by up to 20.5% on the TU-Berlin Extension dataset, and up to 26.4% on the Sketchy dataset respectively.

References

1. Saavedra, J.M., Bustos, B.: An improved histogram of edge local orientations for sketch-based image retrieval. In: Pattern Recognition - 32nd DAGM Symposium, Darmstadt, Germany, September 22-24, 2010. Proceedings. (2010) 432–441
2. Yu, Q., Yang, Y., Liu, F., Song, Y., Xiang, T., Hospedales, T.M.: Sketch-a-net: A deep neural network that beats humans. International Journal of Computer Vision **122**(3) 411–425
3. Bozas, K., Izquierdo, E.: Large scale sketch based image retrieval using patch hashing. In: Advances in Visual Computing - 8th International Symposium, ISVC 2012, Rethymnon, Crete, Greece, July 16-18, 2012, Revised Selected Papers, Part I. (2012) 210–219
4. Yu, Q., Liu, F., Song, Y., Xiang, T., Hospedales, T.M., Loy, C.C.: Sketch me that shoe. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. (2016) 799–807
5. Sangkloy, P., Burnell, N., Ham, C., Hays, J.: The sketchy database: learning to retrieve badly drawn bunnies. ACM Trans. Graph. **35**(4) (2016) 119:1–119:12
6. Song, J., Yu, Q., Song, Y., Xiang, T., Hospedales, T.M.: Deep spatial-semantic attention for fine-grained sketch-based image retrieval. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. (2017) 5552–5561
7. Eitz, M., Hildebrand, K., Boubekeur, T., Alexa, M.: An evaluation of descriptors for large-scale image retrieval from sketched feature lines. Computers & Graphics **34**(5) (2010) 482–498
8. Liu, L., Shen, F., Shen, Y., Liu, X., Shao, L.: Deep sketch hashing: Fast free-hand sketch-based image retrieval. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. (2017) 2298–2307
9. Saavedra, J.M., Barrios, J.M.: Sketch based image retrieval using learned keyshapes (LKS). In: Proceedings of the British Machine Vision Conference 2015, BMVC 2015, Swansea, UK, September 7-10, 2015. (2015) 164.1–164.11
10. Hu, R., Barnard, M., Collomosse, J.P.: Gradient field descriptor for sketch based retrieval and localization. In: Proceedings of the International Conference on Image Processing, ICIP 2010, September 26-29, Hong Kong, China. (2010) 1025–1028
11. Parui, S., Mittal, A.: Similarity-invariant sketch-based image retrieval in large databases. In: Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI. (2014) 398–414
12. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA. (2005) 886–893
13. Hu, R., Collomosse, J.P.: A performance evaluation of gradient field HOG descriptor for sketch based image retrieval. Computer Vision and Image Understanding **117**(7) (2013) 790–806
14. Saavedra, J.M.: Sketch based image retrieval using a soft computation of the histogram of edge local orientations (S-HELO). In: 2014 IEEE International Conference on Image Processing, ICIP 2014, Paris, France, October 27-30, 2014. (2014) 2998–3002
15. Lowe, D.G.: Object recognition from local scale-invariant features. In: ICCV. (1999) 1150–1157

16. Isola, P., Zhu, J., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. (2017) 5967–5976
17. Li, Y., Hospedales, T.M., Song, Y., Gong, S.: Intra-category sketch-based image retrieval by matching deformable part models. In: British Machine Vision Conference, BMVC 2014, Nottingham, UK, September 1-5, 2014. (2014)
18. Liong, V.E., Lu, J., Wang, G., Moulin, P., Zhou, J.: Deep hashing for compact binary codes learning. In: Proc. CVPR. (2015) 2475–2483
19. Liu, W., Mu, C., Kumar, S., Chang, S.F.: Discrete graph hashing. In: Proc. NIPS. (2014) 3419–3427
20. Liu, W., Wang, J., Ji, R., Jiang, Y.G., Chang, S.F.: Supervised hashing with kernels. In: Proc. CVPR. (2012) 2074–2081
21. Shen, F., Shen, C., Liu, W., Shen, H.T.: Supervised discrete hashing. In: Proc. CVPR. (2015) 37–45
22. Weiss, Y., Torralba, A., Fergus, R.: Spectral hashing. In: Proc. NIPS. (2008) 1753–1760
23. Zhang, Z., Chen, Y., Saligrama, V.: Efficient training of very deep neural networks for supervised hashing. In: Proc. CVPR. (2016) 1487–1495
24. Shen, F., Yang, Y., Liu, L., Liu, W., Dacheng Tao, H.T.S.: Asymmetric binary coding for image search. IEEE TMM **19**(9) (2017) 2022–2032
25. Qin, J., Liu, L., Shao, L., Ni, B., Chen, C., Shen, F., Wang, Y.: Binary coding for partial action analysis with limited observation ratios. In: Proc. CVPR. (2017) 146–155
26. Liu, L., Shao, L., Shen, F., Yu, M.: Discretely coding semantic rank orders for image hashing. In: Proc. CVPR. (2017) 1425–1434
27. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. (2014) 2672–2680
28. Denton, E.L., Chintala, S., Fergus, R., et al.: Deep generative image models using a laplacian pyramid of adversarial networks. In: Advances in neural information processing systems. (2015) 1486–1494
29. Mathieu, M.F., Zhao, J.J., Zhao, J., Ramesh, A., Sprechmann, P., LeCun, Y.: Disentangling factors of variation in deep representation using adversarial training. In: Advances in Neural Information Processing Systems. (2016) 5040–5048
30. Sangkloy, P., Lu, J., Fang, C., Yu, F., Hays, J.: Scribbler: Controlling deep image synthesis with sketch and color. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Volume 2. (2017)
31. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. (2017)
32. Zhu, J., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. (2017) 2242–2251
33. Bui, T., Ribeiro, L., Ponti, M., Collomosse, J.P.: Generalisation and sharing in triplet convnets for sketch based visual search. CoRR **abs/1611.05301** (2016)
34. Xu, P., Yin, Q., Huang, Y., Song, Y., Ma, Z., Wang, L., Xiang, T., Kleijn, W.B., Guo, J.: Cross-modal subspace learning for fine-grained sketch-based image retrieval. Neurocomputing **278** (2018) 75–86
35. Li, K., Pang, K., Song, Y., Hospedales, T.M., Zhang, H., Hu, Y.: Fine-grained sketch-based image retrieval: The role of part-aware attributes. In: 2016 IEEE

Winter Conference on Applications of Computer Vision, WACV 2016, Lake Placid, NY, USA, March 7-10, 2016. (2016) 1–9
36. Qi, Y., Song, Y., Zhang, H., Liu, J.: Sketch-based image retrieval via siamese convolutional neural network. In: 2016 IEEE International Conference on Image Processing, ICIP 2016, Phoenix, AZ, USA, September 25-28, 2016. (2016) 2460–2464
37. Xu, P., Yin, Q., Qi, Y., Song, Y., Ma, Z., Wang, L., Guo, J.: Instance-level coupled subspace learning for fine-grained sketch-based image retrieval. In: Computer Vision - ECCV 2016 Workshops - Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part I. (2016) 19–34
38. Li, K., Pang, K., Song, Y., Hospedales, T.M., Xiang, T., Zhang, H.: Synergistic instance-level subspace alignment for fine-grained sketch-based image retrieval. IEEE Trans. Image Processing **26**(12) (2017) 5908–5921
39. Song, J., Qian, Y., Song, Y.Z., Xiang, T., Hospedales, T.: Deep spatial-semantic attention for fine-grained sketch-based image retrieval. In: ICCV. (2017)
40. Zhang, X., Zhou, S., Feng, J., Lai, H., Li, B., Pan, Y., Yin, J., Yan, S.: Hashgan: Attention-aware deep adversarial hashing for cross modal retrieval. CoRR **abs/1711.09347** (2017)
41. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. (2016) 770–778
42. Gui, J., Liu, T., Sun, Z., Tao, D., Tan, T.: Fast supervised discrete hashing. IEEE Trans. Pattern Anal. Mach. Intell. **40**(2) (2018) 490–496
43. Eitz, M., Hays, J., Alexa, M.: How do humans sketch objects? ACM Trans. Graph. **31**(4) (2012) 44:1–44:10
44. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Proc. CVPR. (2009) 248–255
45. Zhang, H., Liu, S., Zhang, C., Ren, W., Wang, R., Cao, X.: Sketchnet: Sketch classification with web images. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. (2016) 1105–1113
46. Wang, F., Kang, L., Li, Y.: Sketch-based 3d shape retrieval using convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015. (2015) 1875–1883
47. Ding, G., Guo, Y., Zhou, J.: Collective matrix factorization hashing for multimodal data. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014. (2014) 2083–2090
48. Bronstein, M.M., Bronstein, A.M., Michel, F., Paragios, N.: Data fusion through cross-modality metric learning using similarity-sensitive hashing. In: The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010. (2010) 3594–3601
49. Kumar, S., Udupa, R.: Learning hash functions for cross-view similarity search. In: IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011. (2011) 1360–1365
50. Zhang, D., Li, W.: Large-scale supervised multimodal hashing with semantic correlation maximization. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada. (2014) 2177–2183
51. Lin, Z., Ding, G., Hu, M., Wang, J.: Semantics-preserving hashing for cross-view retrieval. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015. (2015) 3864–3872

52. Jiang, Q., Li, W.: Deep cross-modal hashing. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. (2017) 3270–3278
53. Vía, J., Santamaría, I., Pérez, J.: Canonical correlation analysis (CCA) algorithms for multiple data sets: Application to blind SIMO equalization. In: 13th European Signal Processing Conference, EUSIPCO 2005, Antalya, Turkey, September 4-8, 2005. (2005) 1–4
54. Liu, H., Ma, Z., Han, J., Chen, Z., Zheng, Z.: Regularized partial least squares for multi-label learning. Int. J. Machine Learning & Cybernetics **9**(2) (2018) 335–346
55. Liao, S., Hu, Y., Zhu, X., Li, S.Z.: Person re-identification by local maximal occurrence representation and metric learning. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015. (2015) 2197–2206
56. Xie, W., Peng, Y., Xiao, J.: Cross-view feature learning for scalable social image analysis. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada. (2014) 201–207
57. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2014)
58. Li, Y., Hospedales, T.M., Song, Y., Gong, S.: Free-hand sketch recognition by multi-kernel feature learning. Computer Vision and Image Understanding **137** (2015) 1–11
59. Zhang, H., Cissé, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. CoRR **abs/1710.09412** (2017)