Deep Triplet Quantization

Bin Liu[1], Yue Cao[1], Mingsheng Long[1](✉), Jianmin Wang[1], and Jingdong Wang[2]
[1]School of Software, Tsinghua University, Beijing 100084, China
[1]Beijing National Research Center for Information Science and Technology
[2]Microsoft Research Asia
{liubinthss,caoyue10}@gmail.com,{mingsheng,jimwang}@tsinghua.edu.cn,jingdw@microsoft.com

ABSTRACT

Deep hashing establishes efficient and effective image retrieval by end-to-end learning of deep representations and hash codes from similarity data. We present a compact coding solution, focusing on deep learning to quantization approach that has shown superior performance over hashing solutions for similarity retrieval. We propose Deep Triplet Quantization (DTQ), a novel approach to learning deep quantization models from the similarity triplets. To enable more effective triplet training, we design a new triplet selection approach, Group Hard, that randomly selects hard triplets in each image group. To generate compact binary codes, we further apply a triplet quantization with weak orthogonality during triplet training. The quantization loss reduces the codebook redundancy and enhances the quantizability of deep representations through back-propagation. Extensive experiments demonstrate that DTQ can generate high-quality and compact binary codes, which yields state-of-the-art image retrieval performance on three benchmark datasets, NUS-WIDE, CIFAR-10, and MS-COCO.

CCS CONCEPTS

• **Information systems → Image search**; • **Computing methodologies → Neural networks**;

KEYWORDS

Deep hashing, Quantization, Image search

ACM Reference Format:
Bin Liu[1], Yue Cao[1], Mingsheng Long[1](✉), Jianmin Wang[1], and Jingdong Wang[2]. 2018. Deep Triplet Quantization. In *2018 ACM Multimedia Conference (MM '18), October 22–26, 2018, Seoul, Republic of Korea.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3240508.3240543

1 INTRODUCTION

Approximate nearest neighbors (ANN) search has been widely applied to retrieve large-scale multimedia data in search engines and social networks. Due to the low storage cost and fast retrieval speed, learning to hash has been increasingly popular in the ANN research community, which transforms high-dimensional media data into compact binary codes and generates similar binary codes for similar data items. This paper will focus on data-dependent hashing schemes for efficient image retrieval, which have achieved better performance than data-independent hashing methods, e.g. Locality-Sensitive Hashing (LSH) [11].

A rich line of hashing methods have been proposed to enable efficient ANN search using Hamming distance [9, 13, 18, 23, 27, 32, 39]. Recently, deep hashing methods [3, 5, 8, 15, 19, 20, 22, 30, 37, 42] have shown that both image representation and hash coding can be learned more effectively using deep neural networks, resulting in state-of-the-art results on many benchmark datasets. In particular, it proves crucial to jointly preserve similarity and control quantization error of converting continuous representations to binary codes [3, 20, 22, 42]. However, a pivotal weakness of these deep hashing methods is that they first learn continuous deep representations, and then convert them into hash codes by a separated binarization step. By *continuous relaxation*, i.e. solving the original discrete optimization of hash codes with continuous optimization, the optimization problem deviates significantly from the original hashing objective. As a result, these methods cannot learn *exactly* compact binary hash codes in their optimization.

To address the limitation of continuous relaxation, Deep Quantization Network (DQN) [2] and Deep Visual-Semantic Quantization (DVSQ) [1] are proposed to integrate quantization method [10, 35, 40] and deep learning. The quantization method represents each point by a short binary code formed by the index of the nearest center, which can generate natively binary codes and empirically achieve better performance than hashing methods for ANN search. However, previous deep quantization methods are either point-wise method that relies on expensive class-label information, or pairwise method that cannot capture the *relative* similarity between images, i.e. a pair of images should not be seen as *absolutely* similar or dissimilar. In other words, there should be a continuous spectrum from very similar to very dissimilar relations.

Recently, the *triplet loss* [28] has been studied for computer vision problems. The triplet loss captures the *relative* similarity, which only brings anchor images closer to positive samples than to negative samples, hence it fits the ranking tasks naturally and achieves better performance than point-wise and pairwise losses for retrieval tasks. However, how to enable effective triplet training for deep learning to quantization with only pairwise similarity available still remains a challenge. Note that, without effective triplet selection, previous deep hashing method with triplet loss [19] cannot achieve superior results. Hence, how to select good triplets for effective training in deep quantization also remains an open problem.

Towards these open problems, this paper presents Deep Triplet Quantization (DTQ) for efficient and effective image retrieval, which introduces a novel triplet training strategy to deep quantization, offering superior retrieval performance. The proposed solution is

MM '18, October 22–26, 2018, Seoul, Republic of Korea

ACM ISBN 978-1-4503-5665-7/18/10…$15.00
https://doi.org/10.1145/3240508.3240543

comprised of four main components: 1) a novel triplet selection module, *Group Hard*, to mine good triplets for effective triplet training; 2) a standard deep convolutional neural network (CNN), e.g. AlexNet or ResNet, for learning deep representations; 3) a well-specified triplet loss for pulling together similar pairs and pushing away dissimilar pairs; and 4) a novel triplet quantization loss with weak orthogonality constraint for converting the deep representations of different samples (such as the anchor, positive and negative samples) in the triplets into B-bit compact binary codes. The weak-orthogonality reduces the redundancy of codebooks and controls the quantizability of deep representations. Comprehensive empirical evidence shows that the proposed DTQ can generate compact binary codes and yield state-of-the-art retrieval results on three image retrieval benchmarks, NUS-WIDE, CIFAR-10, and MS-COCO.

2 RELATED WORK

Existing hashing methods can be categorized into unsupervised hashing and supervised hashing [9, 12, 13, 18, 23, 25, 27, 32, 33, 38, 39]. Please refer to [34] for a comprehensive survey.

Unsupervised hashing methods learn hash functions to encode data points to binary codes by training from unlabeled data. Typical learning criteria include reconstruction error minimization [13, 16, 29] and graph learning [24, 36]. Supervised hashing explores supervised information (e.g. given similarity or relevance feedback) to learn compact hash codes. Binary Reconstruction Embedding (BRE) [18] pursues hash functions by minimizing the squared errors between the distances of data points and the distances of their corresponding hash codes. Minimal Loss Hashing (MLH) [27] and Hamming Distance Metric Learning [28] learn hash codes by minimizing the triplet loss functions based on similarity of data points. Supervised Hashing with Kernels (KSH) [23] and Supervised Discrete Hashing (SDH) [30] build discrete binary codes by minimizing the Hamming distances across similar pairs and maximizing the Hamming distances across dissimilar pairs.

As deep convolutional networks [14, 17] yield advantageous performance on many computer vision tasks, deep hashing methods have attracted wide attention recently. CNNH [37] adopts a two-stage strategy in which the first stage learns binary hash codes and the second stage learns a deep-network based hash function to fit the codes. DNNH [19] improved CNNH with a simultaneous feature learning and hash coding pipeline such that deep representations and hash codes are optimized by the triplet loss. DHN [42] and HashNet [3] improve DNNH by jointly preserving the pairwise semantic similarity and controlling the quantization error by simultaneously optimizing the pairwise cross-entropy loss and quantization loss via a multi-task approach.

Quantization methods [1, 2] represent each point by a short code formed by the index of the nearest center, have been shown to give more powerful representation ability than hashing for approximate nearest neighbor search. To our best knowledge, Deep Quantization Network (DQN) [2] and Deep Visual-Semantic Quantization (DVSQ) [1] are the only two prior works on deep learning to quantization. DQN jointly learns deep representations via a pairwise cosine loss and a product quantization loss [16] for generating compact binary codes. DVSQ proposes a pointwise adaptive-margin Hinge loss exploring class labels, and a visual-semantic quantization loss for inner-product search.

There are several key differences between our work and previous deep learning to quantization methods. 1) Our work introduces a novel triplet training strategy to deep quantization framework for efficient similarity retrieval. It is worth noting that DTQ can learn compact binary codes when only the *relative* similarity information is available, which is more general than the label-based quantization method DVSQ. 2) During the triplet learning procedure, DTQ proposes a novel triplet mining strategy, *Group Hard*, resulting in faster convergence and better search accuracy. 3) DTQ proposes a novel triplet quantization loss with weak orthogonality constraint to reduce coding redundancy. An end-to-end architecture to join the above three terms yield both efficient and effective image retrieval.

3 DEEP TRIPLET QUANTIZATION

In similarity retrieval, we are given N training points $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^N$, where some pairs of points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are given with *pairwise* similarity labels s_{ij}, where $s_{ij} = 1$ if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are similar while $s_{ij} = 0$ if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are dissimilar. The goal of deep learning to quantization is to learn a composite quantizer $q : \boldsymbol{x} \mapsto \boldsymbol{b} \in \{0,1\}^B$ from input space to binary coding space $\{0,1\}^B$ through deep networks, which encodes each point $\boldsymbol{x}$ into B-bit binary code $\boldsymbol{b} = q(\boldsymbol{x})$ such that the supervision in the training data can be maximally preserved. In supervised hashing, the similarity pairs $\{(\boldsymbol{x}_i, \boldsymbol{x}_j, s_{ij}) : s_{ij} \in \mathcal{S}\}$ are readily available from semantic labels or relevance feedbacks from click-through data in many image search engines.

We propose Deep Triplet Quantization (**DTQ**), an end-to-end architecture to join deep learning and quantization, as shown in Figure 1. DTQ has four key components: 1) a novel triplet selection module, *Group Hard*, to mine a appropriate number of good triplets for effective triplet training; 2) a standard deep convolutional neural network (CNN), e.g. AlexNet, VGG, or ResNet, for learning deep representations; 3) a well-specified triplet loss for pulling together similar pairs and pushing away dissimilar pairs; and 4) a novel triplet quantization loss with weak orthogonality constraint for converting deep representations of different samples (the *anchor*, *positive* and *negative* samples) in triplets into B-bit compact binary codes and controlling the quantizability of the deep representations.

3.1 Triplet Training

We train a convolutional network from image triplets $\mathcal{T}=\{\boldsymbol{t}_i\}_{i=1}^{N_t}$. Each triplet $\boldsymbol{t}_i=\langle \boldsymbol{x}_i^a, \boldsymbol{x}_i^p, \boldsymbol{x}_i^n \rangle$ is constructed from pairwise similarity data $\{(\boldsymbol{x}_i, \boldsymbol{x}_j, s_{ij}) : s_{ij} \in \mathcal{S}\}$ as follows: for each *anchor* image $\boldsymbol{x}_i^a$, we find a *positive* image $\boldsymbol{x}_i^p$ with $s_{ap} = 1$ ($\boldsymbol{x}_i^a$ and $\boldsymbol{x}_i^p$ are similar), and a *negative* image $\boldsymbol{x}_i^n$ with $s_{an} = 0$ ($\boldsymbol{x}_i^a$ and $\boldsymbol{x}_i^n$ are dissimilar). Given a triplet $\boldsymbol{t}_i=\langle \boldsymbol{x}_i^a, \boldsymbol{x}_i^p, \boldsymbol{x}_i^n \rangle$, the deep network maps the triplet $\boldsymbol{t}_i$ into a learned feature space with $f(\boldsymbol{t}_i)=\langle z_i^a, z_i^p, z_i^n \rangle$. We ensure that an anchor image $\boldsymbol{x}_i^a$ is closer to all positive images $\boldsymbol{x}_i^p$ than to all negative images $\boldsymbol{x}_i^n$. And the *relative* similarity between the images in triplets, $\boldsymbol{x}_i^a, \boldsymbol{x}_i^p, \boldsymbol{x}_i^n$, are measured by the Euclidean distances between their deep features, z_i^a, z_i^p, z_i^n. Thus the triplet loss is

$$L = \sum_{i=1}^{N_t} L_i = \sum_{i=1}^{N_t} \max\left(0, \delta - \left\| z_i^a - z_i^n \right\|_2^2 + \left\| z_i^a - z_i^p \right\|_2^2\right), \quad (1)$$

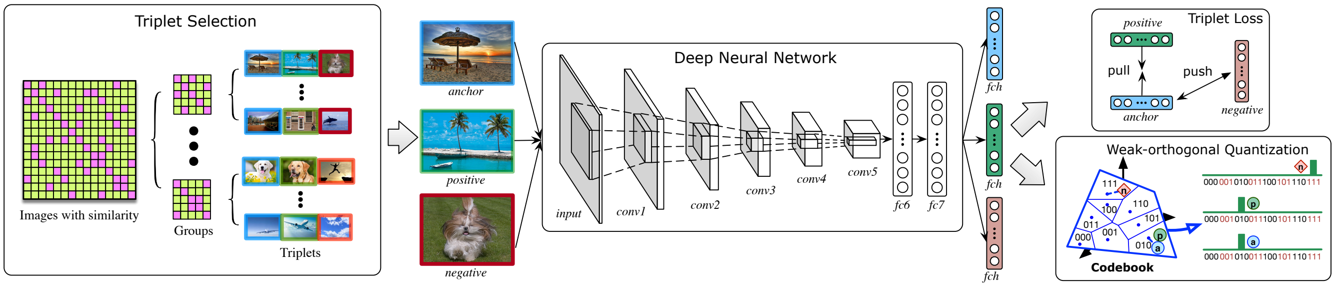


Figure 1: The proposed Deep Triplet Quantization (DTQ) model consists of four main components: 1) a novel triplet selection module, *Group Hard*, to mine good triplets for effective triplet training and faster convergence; 2) a standard deep convolutional neural network (CNN), e.g. AlexNet, VGG or ResNet, for learning deep representations; 3) a well-specified triplet loss for pulling together similar pairs and pushing away dissimilar pairs; and 4) a novel triplet quantization loss with weak orthogonality constraint for converting the deep representations of different samples (the anchor, positive and negative samples) in the triplets into B-bit compact binary codes and controlling the quantizability of the deep representations. *Best viewed in color.*

where δ is a margin that is enforced between positive and negative pairs, and $\mathcal{T}$ is the set of cardinality N_t for all possible triplets in the training set. Compared to the widely-used pointwise and pairwise metric-learning losses [1, 2] in previous deep quantization methods, the triplet loss (1) only requires anchor samples to be more similar to positive samples than to negative samples, by a specifically margin. This establishes a *relative* similarity relation between images, thus is much more reasonable than the *absolute* similarity relation used in previous pointwise or pairwise approaches.

However, as the dataset gets larger, the number of triplets grows cubically, and generating all possible triplets would result in many easy triplets with $L_i = 0$ in Eq. (1), which would not contribute to the training and suffer from slower convergence. Note that, without a sophisticated triplet selection procedure, previous deep hashing methods with the triplet loss [19] cannot achieve superior performance. Consequently, it is crucial to mine good triplets for effective triplet training and faster convergence. In this paper, we propose a novel triplet selection module, *Group Hard*, to ensure the number of mined valid triplets is neither too big nor too small. The core idea is that we first randomly split the training data into several groups $\{G_i\}_{i=1}^{|G|}$, then *randomly* select one *hard* negative sample for each anchor-positive pair in one group. The proposed triplet selection method is formulated as

$$\mathcal{T} = \bigcup_{i=1}^{|G|} \bigcup_{a \in G_i} \bigcup_{p \in G_i^p} \mathrm{rand}\left(G_i^n\right), \tag{2}$$

where $G_i^p = \left\{p \in G_i : p \neq a, s_{ap} = 1\right\}$ is the group of positive samples consisting of the samples similar to the anchor a in the ith group, $\mathrm{rand}(G_i^n)$ is the random function that randomly chooses one negative sample from the group of hard negative samples $G_i^n = \left\{n \in G_i : \delta - \left\|z_i^a - z_i^n\right\|_2^2 + \left\|z_i^a - z_i^p\right\|_2^2 > 0, s_{an} = 0\right\}$. Here *hard* negative sample $\boldsymbol{x}_i^n$ is defined as having non-zero loss value for a triplet $\boldsymbol{t}_i = \langle \boldsymbol{x}_i^a, \boldsymbol{x}_i^p, \boldsymbol{x}_i^n \rangle$. Note that, mining only the triplets with the hardest negative images would select the outliers in the dataset and make it unable to learn ground truth relative similarity. Thus in this paper, the proposed DTQ only selects the negative examples with moderate hardness, based on the random sampling $\mathrm{rand}(G_i^n)$ instead of $\mathrm{argmax}(G_i^n)$ in Eq. (2).

As the training proceeds, the average of triplet loss becomes smaller and the size of the hard triplets reduces. To ensure that there are enough hard triplets each epoch for effective triplet training, we design a decay strategy for the size of groups $|G|$ as: if the actual number of valid hard triplets is lower than the minimum number of the valid hard triplets (the constant MIN_TRIPLETS in Algorithm 1), the size of the groups is halved until $|G| = 1$.

Complexity: Similar to previous work on triplet training [41], we can prune the triplets with zero losses ($L_i = 0$), resulting a valid triplet set $\mathcal{T}$ whose size $|\mathcal{T}|$ is much smaller than the possible number N^3 of triplets. Through the proposed *Group Hard* selection strategy that chooses one negative sample for each anchor-positive pair in each group, the number of the candidate triplets for training is further reduced to $|\mathcal{T}|/|G|$. Furthermore, all the selected triplets are *hard* triplets ($L_i > 0$ in Eq. (1)), and the total amount can be controlled in a suitable range by adjusting the number of groups G, resulting in effective triplet training and higher retrieval accuracy.

3.2 Weak-Orthogonal Quantization

While triplet training with Group Hard selection enables effective image retrieval, efficient image retrieval is enabled by a novel triplet quantization model. As each batch used for training the deep neural networks is comprised of triplets, the proposed quantization model should be compatible with the triplet training. For the ith triplet, each image representation $\boldsymbol{z}_i^*$, where $* \in \{a, p, n\}$, is quantized using a set of M codebooks $\boldsymbol{C}^* = [\boldsymbol{C}_1^*, \ldots, \boldsymbol{C}_M^*]$, where each codebook $\boldsymbol{C}_m^*$ contains K codewords $\boldsymbol{C}_m^* = [\boldsymbol{C}_{m1}^*, \ldots, \boldsymbol{C}_{mK}^*]$, and each codeword $\boldsymbol{C}_{mk}^*$ is a D-dimensional cluster-centroid vector as in K-means. Corresponding to the M codebooks, we partition the binary codewords assignment vector $\boldsymbol{b}_i^*$ into M 1-of-K indicator vectors $\boldsymbol{b}_i^* = [\boldsymbol{b}_{1i}^*; \ldots; \boldsymbol{b}_{Mi}^*]$, and each indicator vector $\boldsymbol{b}_{mi}^*$ indicates which one (and only one) of the K codewords in the mth codebook is used to approximate the ith data point $\boldsymbol{z}_i^*$. To enable knowledge sharing across the anchors, positive and negative samples in the triplets, we propose a *triplet quantization* approach by sharing the codebooks $\{\boldsymbol{C}_m^* = \boldsymbol{C}_m\}_{m=1}^M$ across different samples in all triplets.

To mitigate the degeneration issue of K-means, we further propose a *weak orthogonality* penalty across the M codebooks, which potentially reduces the redundancy of the multiple codebooks and improves the compactness of the binary codes. The proposed triplet quantization model with weak-orthogonal constraint is defined as

$$Q = \sum_{i=1}^{N_t} \sum_{* \in \{a,p,n\}} \left\| \boldsymbol{z}_i^* - \sum_{m=1}^{M} C_m \boldsymbol{b}_{mi}^* \right\|_2^2 + \gamma \sum_{m=1}^{M} \sum_{m'=1}^{M} \left\| C_m^\mathsf{T} C_{m'} - \boldsymbol{I} \right\|_F^2 \tag{3}$$

where $\left\|\boldsymbol{b}_{mi}^*\right\|_0 = 1, \boldsymbol{b}_{mi}^* \in \{0,1\}^K$, $\|\cdot\|_0$ is the ℓ_0-norm that simply counts the number of the vector's nonzero elements, and γ is the hyper-parameter that controls the degree of orthogonality. The ℓ_0 constraint guarantees that only one codeword in each codebook can be activated to approximate the input data, which leads to compact binary codes. The underlying reason of using M codebooks instead of single codebook to approximate each input data point is to further minimize the quantization error, while single codebook yields significantly lossy compression and large performance drop.

3.3 Deep Triplet Quantization

We enable efficient and effective image retrieval in an end-to-end architecture by integrating the triplet training procedure (1), triplet selection module (2) and the weak-orthogonal quantization (3) in a unified deep triplet quantization (DTQ) model as

$$\min_{\Theta, \boldsymbol{C}, \boldsymbol{B}^*} L + \lambda Q, \tag{4}$$

where $\lambda > 0$ is a hyper-parameter between the triplet loss L and the triplet quantization loss Q, and Θ denotes the set of learnable parameters of the deep network. Through joint optimization problem (4), we can learn the binary codes by jointly preserving the similarity via triplet learning procedure and controlling the quantization error of binarizing continuous representations to compact binary codes. A notable advantage of joint optimization is that we can improve the *quantizability* of the learned deep representations $\{\boldsymbol{z}_i^*\}$ such that they can be quantized more effectively by our weak-orthogonal quantizer (3), yielding more accurate binary codes.

Approximate nearest neighbor (ANN) search by maximum inner-product similarity is a powerful tool for quantization methods [7]. Given a database of N binary codes $\{\boldsymbol{b}_n\}_{n=1}^N$, we follow [1, 2] to adopt *Asymmetric Quantizer Distance* (AQD) as the metric, which computes the inner-product similarity between a given query $\boldsymbol{q}$ and the reconstruction of the database point $\boldsymbol{x}_n$ as

$$\mathrm{AQD}\left(\boldsymbol{q}, \boldsymbol{x}_n\right) = \boldsymbol{z}_q^\mathsf{T} \left(\sum_{m=1}^{M} C_m \boldsymbol{b}_{mn} \right), \tag{5}$$

Given query $\boldsymbol{q}$ and the deep representation $\boldsymbol{z}_q$, these inner-products between $\boldsymbol{z}_q$ and all M codebooks $\{C_m\}_{m=1}^M$ and all K possible values of $\boldsymbol{b}_{mn}$ can be pre-computed and stored in a query-specific $M \times K$ lookup table, which is used to compute AQD between the query and all database points, each entails M table lookups and additions and is slightly more costly than computing the Hamming distance.

3.4 Learning Algorithm

The DTQ optimization problem in Equation (4) consists of three sets of variables: deep convolutional neural network parameters Θ, shared codebook $\boldsymbol{C} = [C_1, \ldots, C_M]$, and binary codes $\boldsymbol{B}^*$. We adopt an alternating optimization paradigm [26] which iteratively updates one variable with the remaining variables fixed.

Algorithm 1: Deep Triplet Quantization (DTQ) Training

```
Input: $N$ training images $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^N$;
Input: Similarity pairs $\mathcal{S} = \{s_{ij}\}_{i,j=1}^N$.
for $epoch = 0$ to MAX_EPOCH do
    Run the model to update $\{z_i\}_{i=1}^N$ for $N$ training images
    if $epoch == 0$ then
        Initialize $\boldsymbol{B}$ and $\boldsymbol{C}$ via Product Quantization [16]
    end
    Split the $N$ training images to $N/|G|$ groups randomly
    $\mathcal{T} \leftarrow \emptyset$
    for group $g = 0$ to $N/|G|$ do
        foreach $\boldsymbol{x}^a, \boldsymbol{x}^p \in G_g$, s.t $s_{ap} = 1$ do
            foreach $\boldsymbol{x}^n \in G_g$, s.t $s_{an} = 0$ do
                if $\delta - \|z^a - z^n\|_2^2 + \left\|z^a - z^p\right\|_2^2 > 0$ then
                    // Triplet $< \boldsymbol{x}^a, \boldsymbol{x}^p, \boldsymbol{x}^n >$ is hard
                    $\mathcal{T}_{ap} \leftarrow \mathcal{T}_{ap} \cup \{< \boldsymbol{x}^a, \boldsymbol{x}^p, \boldsymbol{x}^n >\}$
                end
            end
            // Randomly choose a hard negative sample from $\mathcal{T}_{ap}$
            $\mathcal{T} \leftarrow \mathcal{T} \cup \mathrm{rand}(\mathcal{T}_{\mathrm{ap}})$
        end
    end
    for $i = 0$ to $|\mathcal{T}|$/BATCH_SIZE do
        Train the model using the $i$-th batch of triplets
    end
    Update $\boldsymbol{C}$ and $\boldsymbol{B}$ with Eqn. (7) and Eqn. (8) respectively
    if $|\mathcal{T}|$ < MIN_TRIPLETS and $|G| > 1$ then
        // Halve the size of the groups
        $|G| \leftarrow \lfloor \frac{|G|}{2} \rfloor$
    end
end
Output: The trained deep neural networks of DTQ.
```

Learning Θ. The network parameters Θ can be efficiently optimized via standard back-propagation (BP) algorithm. We adopt the automatic differentiation techniques in TensorFlow.

Learning $\boldsymbol{C}$. We update codebook $\boldsymbol{C}$ by rewriting Equation (4) with $\boldsymbol{C}$ as the unknown variables in matrix formulation as follows,

$$\min_{\boldsymbol{C}} \sum_{* \in \{a,p,n\}} \left\|\boldsymbol{Z}^* - \boldsymbol{C}\boldsymbol{B}^*\right\|_F^2 + \gamma \left\|\boldsymbol{C}^\mathsf{T}\boldsymbol{C} - \boldsymbol{I}\right\|_F^2. \tag{6}$$

We adopt the gradient descent to update $\boldsymbol{C}$, $\boldsymbol{C} \leftarrow \boldsymbol{C} - \eta \frac{\partial Q(C)}{\partial C}$, and

$$\frac{\partial Q\left(\boldsymbol{C}\right)}{\partial \boldsymbol{C}} = 2 \sum_{* \in \{a,p,n\}} \boldsymbol{C}\boldsymbol{B}^*\boldsymbol{B}^{*\mathsf{T}} - 2 \sum_{* \in \{a,p,n\}} \boldsymbol{Z}^*\boldsymbol{B}^{*\mathsf{T}} + 2\gamma \boldsymbol{C}\left(2\boldsymbol{C}^\mathsf{T}\boldsymbol{C} - \boldsymbol{I}\right) \tag{7}$$

where η is a learning rate. We can further speed up computation by first solving $\boldsymbol{C}$ with $\gamma = 0$, which leads to an analytic solution $\boldsymbol{C} = \left[\sum_{* \in \{a,p,n\}} \boldsymbol{Z}^*\boldsymbol{B}^{*\mathsf{T}}\right] \left[\sum_{* \in \{a,p,n\}} \boldsymbol{B}^*\boldsymbol{B}^{*\mathsf{T}}\right]^{-1}$, then updating $\boldsymbol{C}$ with this solution as the starting point of gradient descent.

Learning $\boldsymbol{B}$. As each $\boldsymbol{b}_i^*$ is independent on the rest of $\{\boldsymbol{b}_{i'}^*\}_{i' \neq i}$, the optimization for $\boldsymbol{B}^*$ can be decomposed to $3N_t$ subproblems,

$$\begin{aligned} \min_{\boldsymbol{b}_i^*} & \left\| z_i^* - \sum_{m=1}^{M} C_m \boldsymbol{b}_{mi}^* \right\|^2 \\ \text{s.t.} \quad & \left\| \boldsymbol{b}_{mi}^* \right\|_0 = 1, \boldsymbol{b}_{mi}^* \in \{0,1\}^K. \end{aligned} \tag{8}$$

This is essentially a high-order Markov Random Field (MRF) problem. As the MRF problem is generally NP-hard, we resort to the Iterated Conditional Modes (ICM) algorithm [40] that solves M indicators $\{\boldsymbol{b}_{mi}^*\}_{m=1}^M$ alternatively. Specifically, given $\{\boldsymbol{b}_{m'i}^*\}_{m' \neq m}$ fixed, we update $\boldsymbol{b}_{mi}^*$ by exhaustively checking all the codewords in the codebook $\boldsymbol{C}_m$, finding the specific codeword with mimimal objective in (8), and setting the corresponding entry of $\boldsymbol{b}_{mi}^*$ as 1 and the rest as 0. The ICM algorithm is guaranteed to converge to local minima, and can be terminated if maximum iteration is reached. And the training procedure of DTQ is summarized in Algorithm 1.

4 EXPERIMENTS

We conduct extensive experiments to evaluate the efficacy of the proposed DTQ approach against several state-of-the-art shallow and deep hashing methods on three image retrieval benchmark datasets, NUS-WIDE, CIFAR-10, and MS-COCO. Project codes and detailed configurations will be available at https://github.com/thuml.

4.1 Setup

The evaluation is conducted on three widely used image retrieval benchmark xdatasets: NUS-WIDE, CIFAR-10, and MS-COCO.

NUS-WIDE[1] [4] is a public image dataset which contains 269,648 images in 81 ground truth categories. We follow similar experimental protocols in [1, 2], and randomly sample 5,000 images as query points, with the remaining images used as the database and randomly sample 10,000 images from the database for training.

CIFAR-10[2] is a public dataset with 60,000 tiny images in 10 classes. We follow the protocol in [2] to randomly select 100 images per class as the query set, 500 images per class for training, and the rest images as the database.

MS-COCO[3] [21] is a dataset for image recognition, segmentation and captioning. The current release contains 82,783 training images and 40,504 validation images, where each image is labeled by some of the 80 semantic concepts. We randomly sample 5,000 images as the query points, with the rest used as the database, and randomly sample 10,000 images from the database for training.

Following standard evaluation protocol as previous work [1, 3, 19, 37, 42], the similarity information for hash function learning and for ground-truth evaluation is constructed from image labels: if two images i and j share at least one label, they are similar and $s_{ij} = 1$, otherwise they are dissimilar and $s_{ij} = 0$. Though we use the ground truth image labels to construct the similarity information, the proposed DTQ can learn compact binary codes when only the similarity information is available, more general than label-based hashing and quantization methods [1, 2].

We compare the retrieval performance of **DTQ** with ten state-of-the-art hashing methods, including supervised shallow hashing methods **BRE** [18], **ITQ-CCA** [13], **KSH** [23], **SDH** [30] and supervised deep hashing methods **CNNH** [37], **DNNH** [19], **DHN** [42], **DQN** [2], **HashNet** [3], **DVSQ** [1]. We evaluate retrieval quality based on three standard evaluation metrics: Mean Average Precision (**MAP**), Precision-Recall curves (**PR**), and Precision curves with respect to the numbers of top returned samples (**P@N**). To enable a direct comparison to the published results, all methods use identical training and test sets. We follow [1–3] and adopt MAP@5000 for NUS-WIDE dataset, MAP@5000 for MS-COCO dataset, and MAP@54000 for CIFAR-10 dataset.

Our implementation of DTQ is based on **TensorFlow**. For shallow hashing methods, we use as image features the 4096-dimensional DeCAF$_7$ features [6]. For deep hashing methods, we use as input the original images, and adopt *AlexNet* [17] as the backbone architecture. We fine-tune layers *conv1–fc7* copied from the AlexNet model pre-trained on ImageNet and train the last hash layer via back-propagation. As the last layer is trained from scratch, we set its learning rate to be 10 times that of the lower layers. We use mini-batch stochastic gradient descent (SGD) with 0.9 momentum as the solver, and cross-validate the learning rate from 10^{-5} to 10^{-2} with a multiplicative step-size $10^{\frac{1}{2}}$. We fix $K = 256$ codewords for each codebook as [1]. For each point, the binary code for all M codebooks requires $B = M \ \log_2 \ K = 8M$ bits (i.e. M bytes), where we set $M = B/8$ as B is a hyper-parameters. We fix the mini-batch size of triplets as 128 in each iteration and set the initial number of groups as $|G| = 200$ for NUS-WIDE and MS-COCO, and $|G| = 10$ for CIFAR-10. We select the hyper-parameters of the proposed method DTQ and all comparison methods using the three-fold cross-validation.

4.2 Results

The **MAP** results of all methods are listed in Table 1, showing that the proposed DTQ substantially outperforms all the comparison methods. Specifically, compared to SDH [30], the best shallow hashing method with deep features as input, DTQ achieves absolute increases of **11.1%**, **33.0%** and **20.7%** in the average MAP on NUS-WIDE, CIFAR-10, and MS-COCO respectively. Compared to DVSQ [1], the state-of-the-art deep quantization method with class labels as supervised information, DTQ outperforms DVSQ by large margins of **0.8%**, **6.2%** and **4.9%** in average MAP on the three datasets, NUS-WIDE, CIFAR-10, and MS-COCO, respectively.

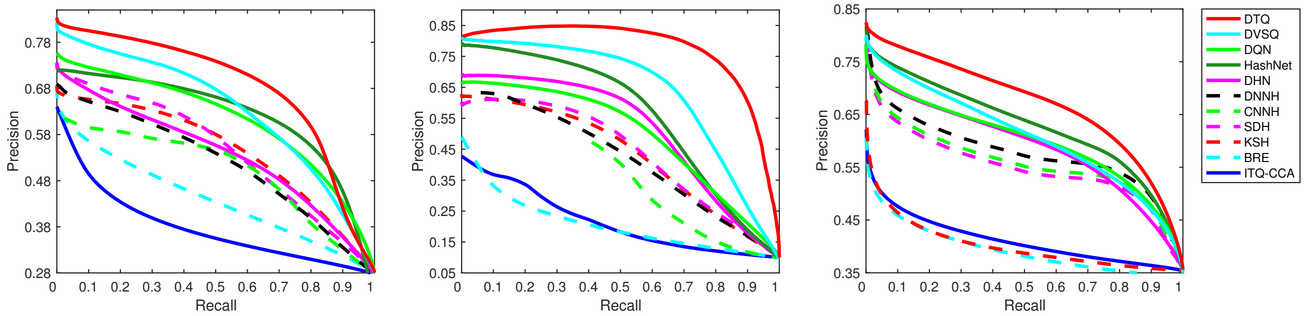
The MAP results reveal several interesting insights. **1)** Shallow hashing methods cannot learn discriminative deep representations and hash codes through end-to-end framework, which explains the fact that they are surpassed by deep hashing methods. **2)** Deep quantization methods DQN and DVSQ learn less lossy binary codes by jointly preserving similarity information and controlling the quantization error, significantly outperforming pioneering methods CNNH and DNNH without reducing the quantization error.

The proposed DTQ improves substantially from the state-of-the-art DVSQ by three important perspectives: **1)** DTQ introduces a novel triplet training strategy to deep quantization framework for efficient similarity retrieval. It is worth noting that DTQ can learn compact binary codes when only the similarity information is available, which is more general than the label-based hashing

[1] http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm
[2] http://www.cs.toronto.edu/kriz/cifar.html
[3] http://mscoco.org

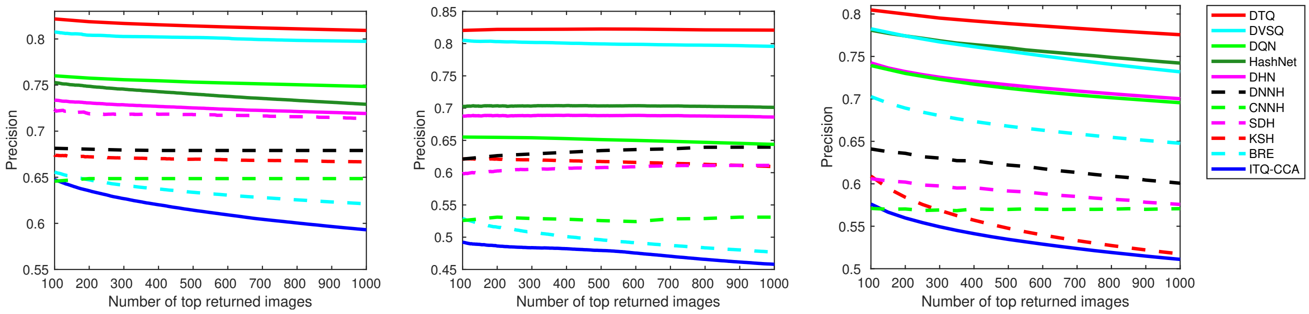
Table 1: Mean Average Precision (MAP) Results for Different Number of Bits on the Three Benchmark Image Datasets

Method	NUS-WIDE				CIFAR-10				MS-COCO			
	8 bits	16 bits	24 bits	32 bits	8 bits	16 bits	24 bits	32 bits	8 bits	16 bits	24 bits	32 bits
ITQ-CCA	0.526	0.575	0.572	0.594	0.315	0.354	0.371	0.414	0.501	0.566	0.563	0.562
BRE	0.550	0.607	0.605	0.608	0.306	0.370	0.428	0.438	0.535	0.592	0.611	0.622
KSH	0.618	0.651	0.672	0.682	0.489	0.524	0.534	0.558	0.492	0.521	0.533	0.534
SDH	0.645	0.688	0.704	0.711	0.356	0.461	0.496	0.520	0.541	0.555	0.560	0.564
CNNH	0.586	0.609	0.628	0.635	0.461	0.476	0.476	0.472	0.505	0.564	0.569	0.574
DNNH	0.638	0.652	0.667	0.687	0.525	0.559	0.566	0.558	0.551	0.593	0.601	0.603
DHN	0.668	0.702	0.713	0.716	0.512	0.568	0.594	0.603	0.607	0.677	0.697	0.701
HashNet	0.613	0.662	0.687	0.699	0.621	0.643	0.660	0.667	0.625	0.687	0.699	0.718
DQN	0.721	0.735	0.747	0.752	0.527	0.551	0.558	0.564	0.649	0.653	0.666	0.685
DVSQ	0.780	0.790	0.792	0.797	0.715	0.727	0.730	0.733	0.704	0.712	0.717	0.720
DTQ	**0.795**	**0.798**	**0.799**	**0.801**	**0.785**	**0.789**	**0.790**	**0.792**	**0.758**	**0.760**	**0.764**	**0.767**



(a) NUS-WIDE (b) CIFAR-10 (c) MS-COCO

Figure 2: Precision-recall curves on the NUS-WIDE, CIFAR-10 and MS-COCO datasets with binary codes @ 32 bits.



(a) NUS-WIDE (b) CIFAR-10 (c) MS-COCO

Figure 3: Precision@top-N curves on the NUS-WIDE, CIFAR-10 and MS-COCO datasets with binary codes @ 32 bits.

method DVSQ. **2)** During the learning of triplet loss, DTQ adopts a novel triplet mining strategy, *Group Hard*, that mines appropriate amount of good triplets for each epoch, resulting in effective triplet training and better performance. **3)** DTQ is the first method to apply weak-orthogonal quantization during triplet training. And back-propagating the triplet quantization loss can remarkably enhance the quantizability of the deep representations.

The retrieval performance in terms of Precision-Recall curves (PR) and Precision curves with respect to different numbers of top returned samples (**P@N**) are shown in Figures 2 and 3, respectively. These metrics are widely used in deploying practical systems. The proposed DTQ significantly outperforms all the comparison methods by large margins under these two evaluation metrics. In particular, DTQ achieves much higher precision at lower recall levels or smaller number of top samples than all compared baselines. This is very desirable for precision-oriented retrieval, where people count more on the top-N returned results with a small N. This justifies the value of our model for practical retrieval systems.

Table 2: Mean Average Precision (MAP) Results of DTQ and Its Variants DTQ-H, DTQ-T, DTQ-2, and DTQ

Method	NUS-WIDE				CIFAR-10				MS-COCO			
	8 bits	16 bits	24 bits	32 bits	8 bits	16 bits	24 bits	32 bits	8 bits	16 bits	24 bits	32 bits
DTQ-H	0.753	0.758	0.763	0.769	0.741	0.747	0.751	0.754	0.708	0.714	0.722	0.729
DTQ-T	0.719	0.722	0.727	0.731	0.663	0.670	0.672	0.679	0.714	0.720	0.728	0.734
DTQ-2	0.752	0.757	0.761	0.768	0.718	0.722	0.726	0.731	0.717	0.725	0.733	0.739
DTQ-Q	0.769	0.773	0.777	0.781	0.750	0.761	0.763	0.765	0.721	0.727	0.734	0.740
DTQ-O	0.785	0.787	0.780	0.788	0.771	0.777	0.779	0.781	0.739	0.745	0.750	0.758
DTQ	**0.795**	**0.798**	**0.799**	**0.801**	**0.785**	**0.789**	**0.790**	**0.792**	**0.758**	**0.760**	**0.764**	**0.767**

4.3 Analysis

4.3.1 Ablation Study. We investigate five variants of DTQ: **1) DTQ-T** is the DTQ variant by replacing the triplet loss in (1) with the widely-used pairwise cross-entropy loss [3, 42]; **2) DTQ-H** is the DTQ variant without Group Hard to mine appropriate amount of good triplets for each epoch during the learning of the triplet loss as [19]; **3) DTQ-2** is the two-step variant of DTQ which first learns the deep representations for all images and then generates compact binary codes via the weak-orthogonal quantization. **4) DTQ-Q** is the DTQ variant which replaces the proposed Triplet Quantization to the Product Quantization [16] used in DQN [2]. **5) DTQ-O** is the DTQ variant by removing the weak orthogonality penalty for redundancy reduction, i.e. $\gamma = 0$.

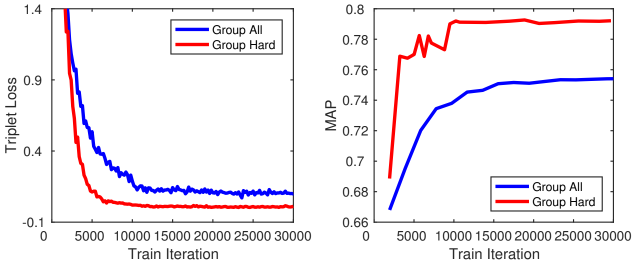
The MAP results for DTQ and it's five variants with respect to different code lengths on three benchmark datasets, NUS-WIDE, CIFAR-10, and MS-COCO are reported in Table 2.

Triplet Loss. DTQ outperforms DTQ-T by very large margins of 7.4%, 11.8% and 3.8% in the average MAP on the three datasets, NUS-WIDE, CIFAR-10, and MS-COCO, respectively. DTQ-T uses the widely-used pairwise cross-entropy loss [3, 42] which achieves state-of-the-art results on previous similarity retrieval tasks. It is worth noting that the triplet loss is a learning to rank method, and tries to bring the anchor and the positive samples closer while also pushing away the negative samples. The DTQ with triplet loss is actually more suitable for the similarity retrieval tasks and naturally gives rise to much better performance than DTQ-T.

Quantizability. Another observation is that by jointly preserving similarity information in the deep representations of image triplets as well as controlling the quantization error of compact binary codes, DTQ outperforms DTQ-2 by 3.9%, 6.4% and 3.4% in the average MAP on the three datasets, NUS-WIDE, CIFAR-10, and MS-COCO. This shows that end-to-end quantization can improve the quantizability of deep feature representations and satisfactorily yield much more accurate retrieval results.

Triplet Quantization. After replacing the proposed Triplet Quantization to Product Quantization [16] used in DQN [2], DTQ-Q yields significantly lossy compression and incur remarkable performance drop of 2.3%, 2.9%, 3.2% in the average MAP on the three datasets, NUS-WIDE, CIFAR-10, and MS-COCO datasets respectively. This proves that the proposed Triplet Quantization with weak orthogonality can effectively learn compact binary codes and enable more effective retrieval than Product Quantization.

Weak-Orthogonal Quantization. Finally, by removing the weak orthogonality penalty, DTQ-O incurs performance drop of 1.3%, 1.2%, 1.4% in the average MAP on the three datasets, NUS-WIDE, CIFAR-10, and MS-COCO datasets respectively. This proves the importance of removing the codebook redundancy and improving the compactness of binary codes for efficient image retrieval.



(a) Loss

(b) mAP

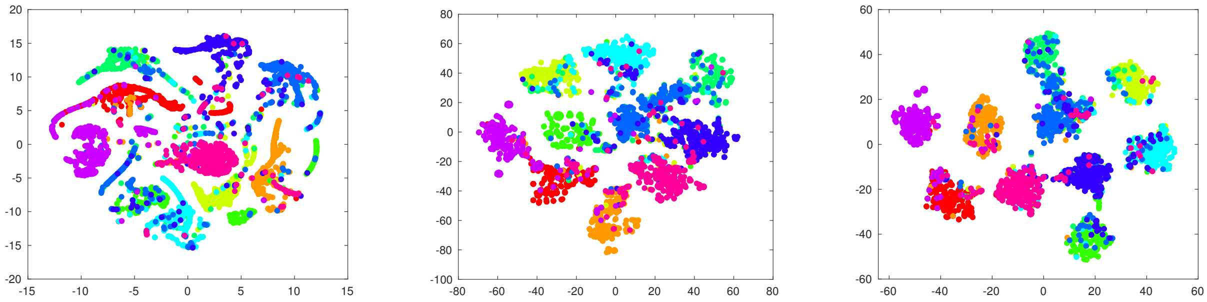
Figure 4: Triplet Loss and MAP curves w.r.t. #iterations.

4.3.2 Triplet Selection. By using the proposed triplet mining strategy, *Group Hard*, DTQ outperforms DTQ-H by large margins of 3.8%, 4.0% and 4.4% in the average MAP on three benchmark datasets, NUS-WIDE, CIFAR-10, and MS-COCO, respectively. As shown in Figure 4, without mining the appropriate amount of hard triplets, the *Group All* training of triplet loss will quickly stagnate, leading to suboptimal convergence quality and MAP results. The proposed triplet mining strategy, *Group Hard*, randomly samples proper amount of useful triplets with hard examples from several randomly partitioned group, resulting in effective training and faster convergence as well as more accurate retrieval performance.

Table 3: MAP on CIFAR-10 for Different Number of Bits

Method	8 bits	16 bits	24 bits	32 bits
DTQ-online	0.703	0.708	0.710	0.713
DTQ	**0.785**	**0.789**	**0.790**	**0.792**

Online Selection. Selecting all batch samples as negative is also known as *online* triplet selection in the literature. Here we conduct a new experiment which uses online triplet selection and selects all hard negative samples in a batch (samples per batch = 192) for each anchor-positive pair. The results are reported in Table 3. Due to the low ratio of the valid hard triplets in each batch for triplet training, DTQ-online (with online triplet selection) fails to achieve satisfactory retrieval results compared with the proposed DTQ.



(a) DVSQ (b) DTQ-2 (c) DTQ

Figure 5: The t-SNE visualizations of deep representations learned by DVSQ, DTQ-2, and DTQ on CIFAR-10 dataset respectively.

As online triplet selection cannot achieve satisfactory results, we adopt *offline* triplet selection, which selects the valid hard triplets at the beginning of each epoch. However, the offline strategy may generate too many candidate triplets and need a huge number of batches per epoch, leading to hard triplets *outdated* for training and potentially wasting most batches of each epoch. To alleviate the outdated effect of hard triplets in offline selection, we split the data into specific groups and select hard triplets within each group, reducing the training triplets from $|\mathcal{T}|$ to $|\mathcal{T}|/|G|$.

We conduct an experiment to count the number of outdated hard triplets during training, shown in Figure 6. By splitting training data into $|G|$ specific groups, the number of outdated hard triplets is significantly reduced, leading to much better MAP results than the original offline triplet selection (i.e. $|G| = 1$). This validates the effectiveness of the proposed offline selection strategy, *Group Hard*.

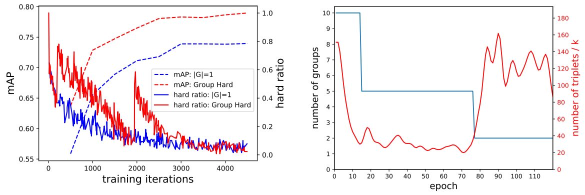


Figure 6: (left) mAP and ratio of non-outdated hard triplets w.r.t. iterations; (right) #groups and #triplets w.r.t. #epochs.

4.3.3 Visualization. We show t-SNE visualization of binary codes and the illustration of top 10 returned images for better understanding the impressive performance improvement of DTQ.

Visualization of Representations. Figure 5 shows the t-SNE visualizations [31] of the deep representations learned by DVSQ [1], DTQ-2, and DTQ on CIFAR-10 dataset. The deep representations of the proposed DTQ exhibit clear discriminative structures with data points in different categories well separated, while the deep representations by DVSQ [1] exhibit relative vague structures. This validates that by introducing the triplet training to deep quantization, the deep representations generated by our DTQ are more discriminative than that generated by DVSQ, enabling more accurate image retrieval. Also, the deep representations of DTQ are more discriminative than that of the two-step variant DTQ-2, showing the efficacy of jointly preserving similarity information in the deep representations of image triplets and controlling the quantization error of compact binary codes via back-propagation.

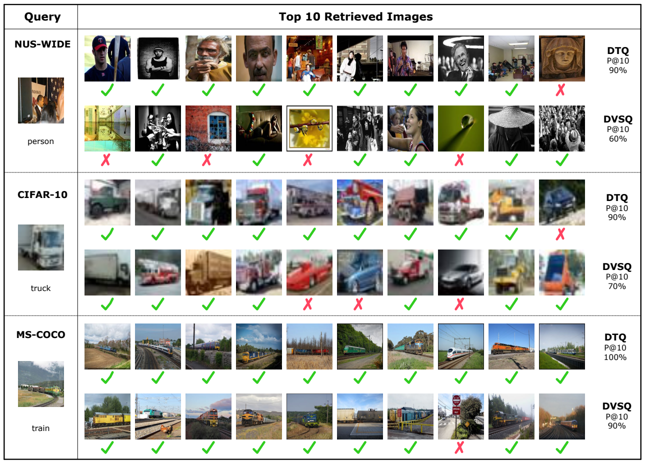


Figure 7: The top 10 images returned by DVSQ and DTQ.

Illustration of Top 10 Results. Figure 7 illustrates the top 10 returned images of DTQ and the best deep hashing baseline DVSQ [1] for three query images on the three datasets NUS-WIDE, CIFAR-10, and MS-COCO, respectively. DTQ yields much more relevant and user-desired retrieval results than the state-of-the-art method.

5 CONCLUSION

This paper proposed Deep Triplet Quantization (DTQ) for efficient image retrieval, which introduces a triplet training strategy to deep quantization framework. Through a novel triplet selection module, Group Hard, an appropriate number of hard triplets are selected for effective triplet training and faster convergence. To enable efficient image retrieval, DTQ can learn compact binary codes by jointly optimizing a novel triplet quantization loss with weak orthogonality. Comprehensive experiments justify that DTQ generates compact binary encoding and yields state-of-the-art retrieval performance on three benchmark datasets NUS-WIDE, CIFAR-10, and MS-COCO.

6 ACKNOWLEDGEMENTS

This work is supported by National Key R&D Program of China (2016YFB1000701), and NSFC grants (61772299, 61672313, 71690231).

REFERENCES

[1] Yue Cao, Mingsheng Long, Jianmin Wang, and Shichen Liu. 2017. Deep visual-semantic quantization for efficient image retrieval. In *CVPR*.

[2] Yue Cao, Mingsheng Long, Jianmin Wang, Han Zhu, and Qingfu Wen. 2016. Deep Quantization Network for Efficient Image Retrieval. AAAI.

[3] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Philip S Yu. 2017. HashNet: Deep Learning to Hash by Continuation. *ICCV* (2017).

[4] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yan-Tao Zheng. 2009. NUS-WIDE: A Real-World Web Image Database from National University of Singapore. In *ICMR*. ACM.

[5] Thanh-Toan Do, Anh-Dzung Doan, and Ngai-Man Cheung. 2016. Learning to hash with binary deep neural network. In *ECCV*. Springer.

[6] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. 2014. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. In *ICML*.

[7] Chao Du and Jingdong Wang. 2014. Inner Product Similarity Search using Compositional Codes. *CoRR* abs/1406.4966 (2014).

[8] Venice Erin Liong, Jiwen Lu, Gang Wang, Pierre Moulin, and Jie Zhou. 2015. Deep Hashing for Compact Binary Codes Learning. In *CVPR*. IEEE.

[9] D. J. Fleet, A. Punjani, and M. Norouzi. 2012. Fast search in Hamming space with multi-index hashing, In CVPR. *CVPR*.

[10] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. 2014. Optimized Product Quantization. *TPAMI* (2014).

[11] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. 1999. Similarity search in high dimensions via hashing. In *VLDB*, Vol. 99. ACM.

[12] Yunchao Gong, Sudhakar Kumar, Henry Rowley, Svetlana Lazebnik, et al. 2013. Learning binary codes for high-dimensional data using bilinear projections. In *CVPR*. IEEE, 484–491.

[13] Yunchao Gong and Svetlana Lazebnik. 2011. Iterative quantization: A procrustean approach to learning binary codes. In *CVPR*. 817–824.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. *CVPR* (2016).

[15] Himalaya Jain, Joaquin Zepeda, Patrick Pérez, and Rémi Gribonval. 2017. SUBIC: A supervised, structured binary code for image search. In *ICCV*.

[16] H. Jegou, M. Douze, and C. Schmid. 2011. Product Quantization for Nearest Neighbor Search. *TPAMI* 33, 1 (Jan 2011), 117–128.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*.

[18] Brian Kulis and Trevor Darrell. 2009. Learning to hash with binary reconstructive embeddings. In *NIPS*. 1042–1050.

[19] Hanjiang Lai, Yan Pan, Ye Liu, and Shuicheng Yan. 2015. Simultaneous Feature Learning and Hash Coding with Deep Neural Networks. In *CVPR*. IEEE.

[20] Wu-Jun Li, Sheng Wang, and Wang-Cheng Kang. 2016. Feature learning based deep supervised hashing with pairwise labels. In *IJCAI*.

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *ECCV*. Springer, 740–755.

[22] Haomiao Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. 2016. Deep supervised hashing for fast image retrieval. In *CVPR*. 2064–2072.

[23] Wei Liu, Jun Wang, Rongrong Ji, Yu-Gang Jiang, and Shih-Fu Chang. 2012. Supervised hashing with kernels. In *CVPR*. IEEE.

[24] Wei Liu, Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. 2011. Hashing with Graphs. In *ICML*. ACM.

[25] Xianglong Liu, Junfeng He, Bo Lang, and Shih-Fu Chang. 2013. Hash bit selection: a unified solution for selection problems in hashing. In *CVPR*. IEEE.

[26] Mingsheng Long, Yue Cao, Jianmin Wang, and Philip S. Yu. 2016. Composite Correlation Quantization for Efficient Multimodal Retrieval. In *SIGIR*.

[27] Mohammad Norouzi and David M Blei. 2011. Minimal loss hashing for compact binary codes. In *ICML*. ACM, 353–360.

[28] Mohammad Norouzi, David M Blei, and Ruslan R Salakhutdinov. 2012. Hamming distance metric learning. In *NIPS*. 1061–1069.

[29] Ruslan Salakhutdinov and Geoffrey E Hinton. 2007. Learning a nonlinear embedding by preserving class neighbourhood structure. In *AISTATS*. 412–419.

[30] Fumin Shen, Chunhua Shen, Wei Liu, and Heng Tao Shen. 2015. Supervised Discrete Hashing. In *CVPR*. IEEE.

[31] L.J.P van der Maaten and G.E. Hinton. Nov 2008. Visualizing High-Dimensional Data Using t-SNE. *JMLR* 9: 2579–2605 (Nov 2008).

[32] Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. 2012. Semi-supervised hashing for large-scale search. *TPAMI* 34, 12 (2012), 2393–2406.

[33] Jianfeng Wang, Jingdong Wang, Jingkuan Song, Xin-Shun Xu, Heng Tao Shen, and Shipeng Li. 2015. Optimized Cartesian K-Means. *IEEE Trans. Knowl. Data Eng.* 27, 1 (2015), 180–192. https://doi.org/10.1109/TKDE.2014.2324592

[34] Jingdong Wang, Ting Zhang, Jingkuan Song, Nicu Sebe, and Heng Tao Shen. 2018. A Survey on Learning to Hash. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 4 (Feb. 2018), 769–790.

[35] Xiaojuan Wang, Ting Zhang, Guo-Jun Qi, Jinhui Tang, and Jingdong Wang. 2016. Supervised quantization for similarity search. In *CVPR*.

[36] Yair Weiss, Antonio Torralba, and Rob Fergus. 2009. Spectral Hashing. In *NIPS*.

[37] Rongkai Xia, Yan Pan, Hanjiang Lai, Cong Liu, and Shuicheng Yan. 2014. Supervised hashing for image retrieval via image representation learning. In *AAAI*.

[38] Felix X Yu, Sanjiv Kumar, Yunchao Gong, and Shih-Fu Chang. 2014. Circulant binary embedding. In *ICML*. ACM, 353–360.

[39] Peichao Zhang, Wei Zhang, Wu-Jun Li, and Minyi Guo. 2014. Supervised hashing with latent factor models. In *SIGIR*. ACM, 173–182.

[40] Ting Zhang, Chao Du, and Jingdong Wang. 2014. Composite Quantization for Approximate Nearest Neighbor Search. In *ICML*. ACM.

[41] Liming Zhao, Xi Li, Jingdong Wang, and Yueting Zhuang. 2017. Deeply-learned part-aligned representations for person re-identification. In *ICCV*.

[42] Han Zhu, Mingsheng Long, Jianmin Wang, and Yue Cao. 2016. Deep Hashing Network for Efficient Similarity Retrieval. AAAI.