Deep Visual-Semantic Quantization for Efficient Image Retrieval*

Yue Cao, Mingsheng Long, Jianmin Wang, and Shichen Liu
KLiss, MOE; TNList; NEL-BDSS; School of Software, Tsinghua University, China
{caoyue10,liushichen95}@gmail.com, {mingsheng,jimwang}@tsinghua.edu.cn

Abstract

Compact coding has been widely applied to approximate nearest neighbor search for large-scale image retrieval, due to its computation efficiency and retrieval quality. This paper presents a compact coding solution with a focus on the deep learning to quantization approach, which improves retrieval quality by end-to-end representation learning and compact encoding and has already shown the superior performance over the hashing solutions for similarity retrieval. We propose Deep Visual-Semantic Quantization (DVSQ), which is the first approach to learning deep quantization models from labeled image data as well as the semantic information underlying general text domains. The main contribution lies in jointly learning deep visual-semantic embeddings and visual-semantic quantizers using carefully-designed hybrid networks and well-specified loss functions. DVSQ enables efficient and effective image retrieval by supporting maximum inner-product search, which is computed based on learned codebooks with fast distance table lookup. Comprehensive empirical evidence shows that DVSQ can generate compact binary codes and yield state-of-the-art similarity retrieval performance on standard benchmarks.

1. Introduction

In the big data era, large-scale and high-dimensional media data has been pervasive in search engines and social networks. To guarantee retrieval quality and computation efficiency, approximate nearest neighbor (ANN) search has attracted increasing attention. Parallel to the traditional indexing methods [24], another advantageous solution is hashing methods [38], which transform high-dimensional data into compact binary codes and generate similar binary codes for similar data items. In this paper, we will focus on learning to quantization methods [38] that build data-dependent binary encoding schemes for efficient image retrieval, which have shown better performance than data-independent hashing methods, e.g. Locality-Sensitive Hashing (LSH) [16].

Many hashing methods have been proposed to enable efficient ANN search of high-dimensional data by ranking the Hamming distance across compact binary hash codes based on shallow architectures [22, 18, 31, 13, 26, 37, 28, 17, 42, 43]. Recently, deep hashing methods [41, 23, 35, 12, 45, 5, 7, 6] have shown that end-to-end learning of feature representation and hash coding can be more effective using deep networks [21, 3], which can naturally encode any nonlinear hash functions. These deep hashing methods have shown state-of-the-art performance on many benchmarks. It is crucial to jointly learn similarity-preserving representations and control quantization error of binarizing continuous representations to binary codes [45]. However, a key disadvantage of these deep hashing methods is that they need to first learn continuous deep representations, which are then converted into hash codes by a separated binarization step. By *continuous relaxation*, i.e. solving the discrete optimization of hash codes with continuous optimization, all these methods essentially solve an optimization problem that deviates significantly from the hashing objective as they cannot learn *exactly* binary hash codes in their alternative optimization.

To address the limitation of continuous relaxation, Cao et al. proposed Deep Quantization Network (DQN) [8] to integrate quantization method [15, 44, 39, 29] and deep learning. The quantization method represents each point by a short code formed by the index of the nearest center, which can generate natively binary codes that yield more powerful representation ability than hashing for approximate nearest neighbor search. DQN gives state-of-the-art image retrieval performance, but several key problems remain unaddressed. First, the semantic space spanned by the image labeling may not be uniformly distributed. As can be observed in Figure 1, some labels (e.g. *cat* and *dog*) may be more similar than others (e.g. *cat* and *airplane*). Such semantic relationship between labels proves to be important for image recognition [14], but has not been explored by prior hashing methods. Second, maximum inner-product search has been prevalent in many online search engines, but it remains unclear how to exploit both visual and semantic information to enable it.

This paper presents Deep Visual-Semantic Quantization (**DVSQ**) for fast image retrieval, which learns end-to-end: (1) a deep visual-semantic embedding model that enables transfer of semantic knowledge extracted from general text

*Contact author: Mingsheng Long (mingsheng@tsinghua.edu.cn).

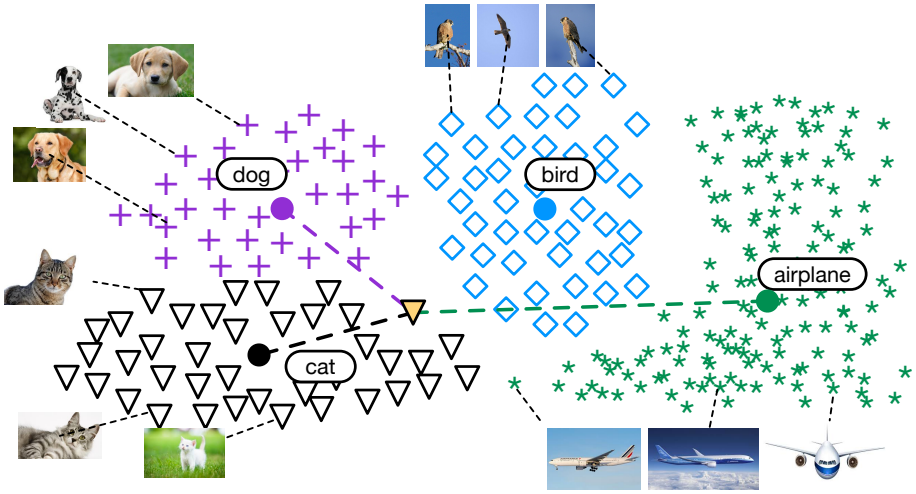


Figure 1. Overview of DVSQ, which jointly maps images to the semantic space of label embeddings and learns visual-semantic quantizers.

domains to the visual retrieval model, (2) a visual-semantic quantization model for fast inner-product search by exploiting both visual and semantic information. The architecture is comprised of four key components: (1) Standard convolutional neural network (CNN), e.g. AlexNet, for learning deep image representations and standard skip-gram model, e.g. Word2Vec, for learning word embeddings, (2) a fully-connected transform layer for transforming the deep image representations into the semantic space spanned by label embeddings, (3) a novel adaptive margin loss for similarity-preserving learning from both image representations and label embeddings and (4) a novel visual-semantic quantization model for converting the image representations into compact binary codes in the semantic space by minimizing quantization error of approximate inner-product search. Comprehensive empirical evidence shows that DVSQ can generate compact binary codes and yield state-of-the-art similarity retrieval performance on standard image datasets.

2. Related Work

Existing hashing methods [22, 18, 31, 13, 26, 37, 28, 17, 42, 43] consist of unsupervised hashing and supervised hashing. Please refer to [38] for a comprehensive survey.

Unsupervised hashing methods learn hash functions that encode data points to binary codes by training from unlabeled data. Typical learning criteria include reconstruction error minimization [34, 18, 20] and graph learning[40, 27]. Supervised hashing explores supervised information (e.g. pairwise similarity or relevance feedback) to learn compact hash codes. Binary Reconstruction Embedding (BRE) [22] 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) [31] and Hamming Distance Metric Learning [32] learn hash codes by minimizing hinge-like loss functions based on similarity of data points. Supervised Hashing with Kernels (KSH) [26] and Supervised Discrete Hashing (SDH) [35] build nonlinear or discrete binary hash codes by minimizing the Hamming distances across similar pairs and maximizing the Hamming distances across dissimilar pairs.

As deep convolutional neural networks (CNNs) [21, 19] yield breakthrough performance on many computer vision tasks, deep learning to hash has attracted attention recently. CNNH [41] adopts a two-stage strategy in which the first stage learns hash codes and the second stage learns a deep-network based hash function to fit the codes. DNNH [23] improved the two-stage CNNH with a simultaneous feature learning and hash coding pipeline such that representations and hash codes can be optimized in a joint learning process. DHN [45] and DSH [25] improve DNNH by cross-entropy or max-margin loss and quantization loss which jointly preserve pairwise similarity and control the quantization error.

It has been shown that quantization methods [15, 44, 39], which represent each point by a short code formed by the index of the nearest center, giving more powerful representation ability than hashing for approximate nearest neighbor search. To our best knowledge, Deep Quantization Network (DQN) [8] is the only prior work on deep learning to quantization. DQN jointly learns a convolutional neural network for image representations and a product quantization [20] for generating compact binary codes. There are several key differences between our work and DQN. (1) Our work can transfer the semantic knowledge learned in general-purpose text domains to enable effective image retrieval by transforming all visual data to the semantic space, while DQN can only use the labeling information. (2) Our work can exploit the nonlinear correlation in the label space, while DQN assumes labels are independent. (3) Our work enables maximum inner-product search by exploiting both visual and semantic information, while DQN only supports minimum Euclidean distance search by mining the visual information.

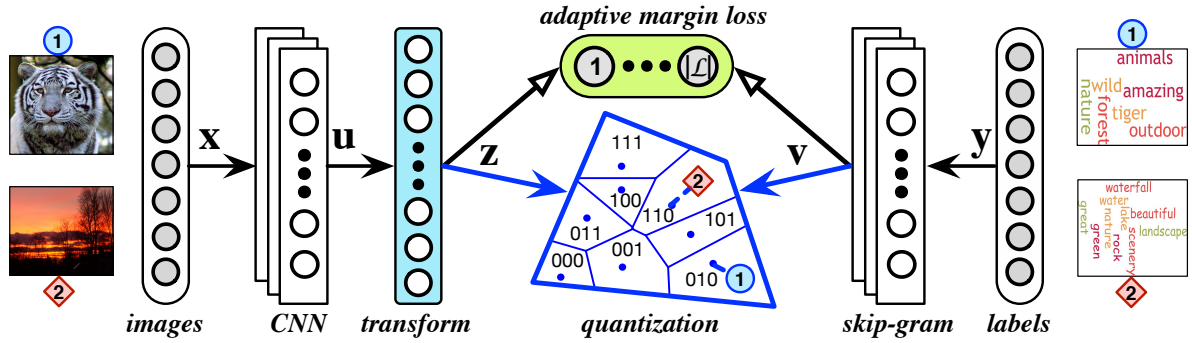


Figure 2. The Deep Visual-Semantic Quantization (DVSQ) for efficient image retrieval, which is comprised of four key components: (1) Standard convolutional neural network (CNN), e.g. AlexNet, for learning deep image representations and standard skip-gram model, e.g. Word2Vec, for learning word embeddings, (2) a fully-connected transform layer for transforming the deep image representations $\{\mathbf{u}\}$ into the semantic space spanned by label embeddings $\{\mathbf{v}\}$, (3) a novel adaptive margin loss for similarity-preserving learning from both image representations $\{\mathbf{u}\}$ and label embeddings $\{\mathbf{v}\}$ and (4) a novel visual-semantic quantization model for converting the image representations into B-bit binary codes in the semantic space by minimizing quantization error of approximate inner-product search. *Best viewed in color.*

3. Deep Visual-Semantic Quantization

In similarity retrieval systems, we are given a training set of N points $\{\mathbf{x}_n, \mathcal{Y}_n\}_{n=1}^N$ and a set of text labels $\mathcal{Y}$, where each point is represented by a P-dimensional feature vector $\mathbf{x}_n \in \mathbb{R}^P$ and is associated with a label set of multiple text labels $\mathcal{Y}_n \subset \mathcal{Y}$. The goal of deep learning to quantization is to learn compositional quantizer $q : \mathbf{x} \mapsto \mathbf{b} \in \{0,1\}^B$ from input space $\mathbb{R}^P$ to binary coding space $\{0,1\}^B$ via deep networks, which encodes each point $\mathbf{x}$ into compact B-bit binary code $\mathbf{b} = q(\mathbf{x})$ such that the supervision in the training data can be preserved in the compact binary codes.

This paper enables efficient image retrieval by presenting a Deep Visual-Semantic Quantization (**DVSQ**) approach as shown in Figure 2, which is an end-to-end deep learning architecture composed of: (1) Standard convolutional neural network (CNN), e.g. AlexNet or GoogLeNet, for learning deep representation $\mathbf{u}$ for each image $\mathbf{x}$ and standard skip-gram model, e.g. Word2Vec, for learning word embedding $\mathbf{v}$ for each text label $y \in \mathcal{Y}$, (2) a fully-connected transform layer for transforming the deep image representations $\{\mathbf{u}\}$ into the semantic space spanned by label embeddings $\{\mathbf{v}\}$, (3) a novel adaptive margin loss for similarity-preserving multi-label learning from both image representations $\{\mathbf{u}\}$ and label embeddings $\{\mathbf{v}\}$, and (4) a novel visual-semantic quantization model for converting the image representations into B-bit binary codes in the semantic space by minimizing quantization error of approximate inner-product search.

3.1. Deep Visual-Semantic Embedding

The DVSQ architecture is composed of a visual model and a text model. We use AlexNet [21] as our visual model, which consists of several convolutional filtering and max-pooling layers, followed by several fully connected layers. We adopt Word2Vec [30] as our text model, which can efficiently learn semantically-meaningful vector representations of words from unannotated text. The model learns to represent each word as a fixed-length embedding vector by predicting adjacent words in the document to give similar embedding vectors for semantically related words. To facilitate fast training, we use the AlexNet model pre-trained on ImageNet 2012 dataset [10], and the Word2Vec model pre-trained on Google News [30], while our DVSQ model is initialized from these pre-trained neural network models.

The goal of DVSQ is to leverage the semantic knowledge learned in text domain, e.g. Google News, and transfer it to our model trained for efficient image retrieval. To this end, we construct deep visual-semantic embedding by taking the image representations $\{\mathbf{u}\}$ of the pre-trained convolutional neural network and re-training them to predict the word embeddings $\{\mathbf{v}\}$ of the image label-text as learned by the skip-gram model. We add a fully connected transform layer with hyperbolic tangent (tanh) activation function to learn a non-linear transformation from deep image representations $\{\mathbf{u}\}$ to the corresponding image embeddings $\{\mathbf{z}\}$ in the semantic space spanned by the label embeddings $\{\mathbf{v}\}$. After transformation, the image embeddings and label embeddings are in the common D-dimensional semantic space and $\mathbf{z}, \mathbf{v} \in \mathbb{R}^D$.

To enable similarity-preserving learning from the given multi-label training data $\{\mathbf{x}_n, \mathcal{Y}_n\}_{n=1}^N$, we propose a novel adaptive margin rank loss for ranking the normalized inner-product similarity between image embeddings $\{\mathbf{z}\}$ and label embeddings $\{\mathbf{v}\}$ such that the DVSQ model produces a higher inner-product similarity between the image embedding and the word embedding of the correct label than between the image embedding and other false text labels. We define the per training example adaptive margin rank loss as

$$L_n = \sum_{i \in \mathcal{Y}_n} \sum_{j \notin \mathcal{Y}_n} \max\left(0, \delta_{ij} - \frac{\mathbf{v}_i^{\mathsf{T}} \mathbf{z}_n}{\|\mathbf{v}_i\| \, \|\mathbf{z}_n\|} + \frac{\mathbf{v}_j^{\mathsf{T}} \mathbf{z}_n}{\|\mathbf{v}_j\| \, \|\mathbf{z}_n\|}\right), \tag{1}$$

where $\mathbf{z}_n$ is the image embedding for image $\mathbf{x}_n$, and $\mathbf{v}_i$ is the word embedding of the correct text label of image $\mathbf{x}_n$ while $\mathbf{v}_j$ is the word embedding of the false text label of image $\mathbf{x}_n$. Note that δ_{ij} denotes the *adaptive* margin that is the key component designed to guarantee that the inner-product similarity between image embedding $\mathbf{z}_n$ and the word embedding of correct text label $\mathbf{v}_i$ should be larger than between false text label $\mathbf{v}_j$ by an adaptive margin. We define the adaptive margin by the inner-product between the word embeddings of correct text label $\mathbf{v}_i$ and false text label $\mathbf{v}_j$,

$$\delta_{ij} = 1 - \frac{\mathbf{v}_i^\mathsf{T}\mathbf{v}_j}{\|\mathbf{v}_i\|\,\|\mathbf{v}_j\|}. \tag{2}$$

The choice of adaptive margin proves to be important. The reason can be intuitively understood in Figure 1. Suppose the yellow triangle denotes our current image $\mathbf{z}_n$, *cat* is the correct label while *dog* and *airplane* are false labels. Since *cat* and *dog* are more similar than *cat* and *airplane* in the semantic space, we can only tolerate smaller margin $\delta_{cat,dog}$ but may require larger margin $\delta_{cat,airplane}$ such that image embeddings are made consistent with the semantic space. It is desirable that the requirement of label-dependent margin can be modeled by the proposed adaptive margin (2) to enable effective learning of deep visual-semantic embeddings.

3.2. Visual-Semantic Inner-Product Quantization

While deep visual-semantic embeddings facilitate effective image retrieval, efficient image retrieval is enabled by a novel visual-semantic inner-product quantization model. Specifically, each image embedding $\mathbf{z}_n$ is quantized using a set of M codebooks $\mathbf{C} = [\mathbf{C}_1, \ldots, \mathbf{C}_M]$, each codebook $\mathbf{C}_m$ contains K codewords $\mathbf{C}_m = [\mathbf{C}_{m1}, \ldots, \mathbf{C}_{mK}]$, and each codeword $\mathbf{C}_{mk}$ is a D-dimensional cluster-centroid vector as kmeans clustering. Corresponding to the M codebooks, we partition the binary codewords assignment vector $\mathbf{b}_n$ into M 1-of-K indicator vectors $\mathbf{b}_n = [\mathbf{b}_{1n}; \ldots; \mathbf{b}_{Mn}]$, and each indicator vector $\mathbf{b}_{mn}$ indicates which one (and only one) of the K codewords in the mth codebook is used to approximate the nth data point. The proposed quantizer encodes each image embedding $\mathbf{z}_n$ as the sum of M codewords, one codeword per codebook, each indicated by the binary assignment vector $\mathbf{b}_n$, that is $\mathbf{z}_n \approx \sum_{m=1}^{M} \mathbf{C}_m\mathbf{b}_{mn}$. Since maximum inner-product search (MIPS) is widely applied in real retrieval systems, we enable MIPS by formulating a visual-semantic inner-product quantization model, with the per training example quantization error defined as

$$Q_n = \sum_{i=1}^{|\mathcal{Y}|} \left(\mathbf{v}_i^\mathsf{T}\mathbf{z}_n - \mathbf{v}_i^\mathsf{T} \sum_{m}^{M} \mathbf{C}_m\mathbf{b}_{mn} \right)^2, \tag{3}$$

which is subject to the discrete constraints $\|\mathbf{b}_{mn}\|_0 = 1$ and $\mathbf{b}_{mn} \in \{0,1\}^K$, with $\|\cdot\|_0$ being the ℓ_0-norm that simply counts the number of the vector's nonzero elements. These constraints guarantee that only one codeword per codebook can be activated to approximate the image embedding $\mathbf{z}_n$, which can generate compact binary codes. The philosophy of using M codebooks instead of single codebook to encode each image is to further minimize the quantization error and generate lossless binary codes with much fewer bits [44, 2].

The quantization loss (3) can be explained as follows: for query $\mathbf{v}_i$ and database image $\mathbf{z}_n$, both embedded into the semantic space by the deep visual-semantic embedding model (1), the inner-product between the query and the database image $\mathbf{v}_i^\mathsf{T}\mathbf{z}_n$, should be as close as possible to the inner-product between the query and the reconstructed database image $\mathbf{v}_i^\mathsf{T}\sum_m^M \mathbf{C}_m\mathbf{b}_{mn}$, which is reduced to minimizing the squared loss between them. A key insight is choosing word embeddings in label set $\mathcal{Y}$ as the query set for training. The reason is that all images are embedded to the semantic space spanned by word embeddings in label set $\mathcal{Y}$, and thus these word embeddings are the most representative queries that can be used to model the underlying query distribution.

3.3. Deep Visual-Semantic Quantization

This paper enables efficient image retrieval in an end-to-end architecture, which learns deep visual-semantic quantization (DVSQ) model by integrating deep visual-semantic embedding model (1) and the visual-semantic inner-product quantization model (3) into a joint optimization problem as

$$\min_{\mathcal{W},\mathbf{C},\mathbf{B}} \sum_{n=1}^{N} (L_n + \lambda Q_n) \tag{4}$$

where $\lambda > 0$ is trade-off between the adaptive margin loss L and the inner-product quantization loss Q, and $\mathcal{W}$ denotes the set of network parameters. Through joint optimization problem (4), we can achieve statistically optimal learning of the binary codes, by jointly transforming the images into the semantic space and controlling the quantization error of binarizing continuous embeddings to compact binary codes. A notable advantage of joint optimization is that we can improve the *quantizability* of the image embeddings $\{\mathbf{z}_n\}$ such that they can be quantized more effectively by our inner-product quantizer (3), yielding more compact binary codes.

Approximate nearest neighbor (ANN) search by maximum inner-product similarity is a powerful tool in quantization methods [11]. Given database binary codes $\{\mathbf{b}_n\}_{n=1}^N$, we use *Asymmetric Quantizer Distance* (AQD) [8] as the metric that computes the inner-product similarity between a given query $\mathbf{q}$ and database point $\mathbf{x}_n$ in the semantic space,

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

Given query $\mathbf{q}$ and embedding $\mathbf{z}_q$ in semantic space, these inner-products between $\mathbf{z}_q$ and all M codebooks $\{\mathbf{C}_m\}_{m=1}^M$

and all K possible values of $\mathbf{b}_{mn}$ 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 DVSQ optimization problem (4) has three sets of variables, the network parameters $\mathcal{W}$, M codebooks $\mathbf{C} = [\mathbf{C}_1, \ldots, \mathbf{C}_M]$, and N binary codes $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_N]$. We adopt an alternating optimization paradigm as [8] which iteratively updates one variable with the rest variables fixed.

Learning $\mathcal{W}$. The network parameters $\mathcal{W}$ can be efficiently optimized via standard back-propagation algorithm by automatic differentiation techniques in TensorFlow [1].

Learning C. We update the M codebooks $\mathbf{C}$ by fixing $\mathcal{W}$ and $\mathbf{B}$ as known variables, and write Equation (4) with $\mathbf{C}$ as unknown variables in matrix formulation as follows,

$$\min_{\mathbf{C}} \operatorname{tr}\left((\mathbf{Z} - \mathbf{CB})^{\mathsf{T}} \Sigma_{\mathbf{V}} (\mathbf{Z} - \mathbf{CB})\right), \tag{6}$$

where $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N]$ is image embeddings matrix, and

$$\Sigma_{\mathbf{V}} = \sum\nolimits_{i=1}^{|\mathcal{Y}|} \mathbf{v}_i \mathbf{v}_i^{\mathsf{T}} \tag{7}$$

is the covariance matrix of the label embeddings $\{\mathbf{v}_i\}_{i=1}^{|\mathcal{Y}|}$, which can reflect the underlying query distributions as all data are embedded to the common semantic space spanned by these label embeddings. Problem (6) is a quadratic problem with analytic solution $\mathbf{C} = \mathbf{ZB}^{\mathsf{T}}(\mathbf{BB}^{\mathsf{T}})^{-1}$. Algorithms such as L-BFGS can speed up the computation.

Learning B. As each $\mathbf{b}_n$ is independent on $\{\mathbf{b}_{n'}\}_{n' \neq n}$, the optimization for $\mathbf{B}$ is decomposed to N subproblems,

$$\min_{\mathbf{b}_n} \left(\mathbf{z}_n - \sum_{m=1}^{M} \mathbf{C}_m \mathbf{b}_{mn}\right)^{\mathsf{T}} \Sigma_{\mathbf{V}} \left(\mathbf{z}_n - \sum_{m=1}^{M} \mathbf{C}_m \mathbf{b}_{mn}\right) \quad \text{s.t.} \quad \|\mathbf{b}_{mn}\|_0 = 1, \mathbf{b}_{mn} \in \{0,1\}^K. \tag{8}$$

This optimization problem is generally NP-hard. It is essentially high-order Markov Random Field (MRF) problem and can be solved by the Iterated Conditional Modes (ICM) algorithm [44, 4] which solves M indicators $\{\mathbf{b}_{mn}\}_{m=1}^{M}$ alternatively. Given $\{\mathbf{b}_{m'n}\}_{m' \neq m}$ fixed, we update $\mathbf{b}_{mn}$ by exhaustively checking all the codeword in codebook $\mathbf{C}_m$, finding the codeword such that the objective in (8) is minimized, and setting the corresponding entry of $\mathbf{b}_{mn}$ as 1 and the rest as 0. The ICM algorithm is guaranteed to converge, and can be terminated if maximum iterations are reached.

4. Experiments

We conduct extensive experiments to evaluate efficiency and effectiveness of DVSQ against several state-of-the-art hashing and quantization methods on three benchmark datasets. Codes and configurations will be available online.

4.1. Setup

We conduct empirical evaluation on three public benchmark datasets, **NUS-WIDE**, **CIFAR-10**, and **ImageNet**.

NUS-WIDE[1] [9] is a public image dataset containing 269,648 images. Each image is annotated by some of the 81 ground truth categories for evaluating retrieval models. We follow similar experimental protocols as DQN [8] and randomly sample 5,000 images as queries, with the remaining images used as the database; furthermore, we randomly sample 10,000 images from the database as training points.

CIFAR-10[2] is a public dataset with 60,000 images in 10 classes. We follow protocol in [8] to randomly select 100 images per class as the query set, 500 images per class as the training set, and the rest images are used as database.

ImageNet[3] is a benchmark dataset for ILSVRC 2015 competition [33]. It contains 1.2M images in the training set and 50K images in the validation set, where each image is single-labeled by one of the 1,000 categories. We randomly select 100 categories, use all the images of these categories in the training set as database, and use all the images in the validation set as queries; furthermore, we randomly select 100 images per category from database for model training.

We follow [23, 8] to evaluate the retrieval quality using three evaluation metrics: Mean Average Precision (MAP), Precision-Recall curves, and Precision curves with respect to the number of top returned results. All methods use identical training and test sets. Given a query, the ground truth is defined as: if a result shares at least one common concept with the query, it is relevant; otherwise it is irrelevant.

We compare the retrieval performance of the proposed **DVSQ** model and its variants with several state-of-the-art hashing methods, including five shallow supervised methods , **SQ** [39], **SDH** [35], **KSH** [26], **BRE** [22] and **ITQ-CCA** [18] as well as five deep supervised methods **DQN** [8], **DSH** [25], **DHN** [45], **DNNH** [23], and **CNNH** [41].

For the deep learning based methods, we directly use the raw image pixels as input. For shallow learning based methods, we use AlexNet [21] to extract 4096-dimensional deep *fc7* features [10] for each image. For the DVSQ approach, we adopt the skip-gram model [30] to construct the semantic space. The skip-gram model was trained on part of the widely used Google News dataset (about 100 billion words) to extract 300-dimensional embedding vector for each label.

Our implementation of DVSQ is based on **TensorFlow**. We adopt AlexNet [21], fine-tune all layers copied from pre-trained model, and train the transform layer via back-propagation. As the transform layer is trained from scratch, we set its learning rate 10 times of the other layers. We use mini-batch SGD with 0.9 momentum, and cross-validate learning rate from 10^{-5} to 10^{-2}. We select parameter λ

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

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

Dataset	NUS-WIDE				CIFAR-10				ImageNet			
	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 [18]	0.526	0.575	0.572	0.594	0.315	0.354	0.371	0.414	0.189	0.270	0.339	0.436
BRE [22]	0.550	0.607	0.605	0.608	0.306	0.370	0.428	0.438	0.251	0.363	0.404	0.453
KSH [26]	0.618	0.651	0.672	0.682	0.489	0.524	0.534	0.558	0.228	0.398	0.499	0.547
SDH [35]	0.645	0.688	0.704	0.711	0.356	0.461	0.496	0.520	0.385	0.516	0.570	0.605
SQ [39]	0.653	0.691	0.698	0.716	0.567	0.583	0.602	0.615	0.465	0.536	0.592	0.611
CNNH [41]	0.586	0.609	0.628	0.635	0.461	0.476	0.476	0.472	0.317	0.402	0.453	0.476
DNNH [23]	0.638	0.652	0.667	0.687	0.525	0.559	0.566	0.558	0.347	0.416	0.497	0.525
DHN [45]	0.668	0.702	0.713	0.716	0.512	0.568	0.594	0.603	0.358	0.426	0.531	0.556
DSH [25]	0.653	0.688	0.695	0.699	0.592	0.625	0.651	0.659	0.332	0.398	0.487	0.537
DQN [8]	0.721	0.735	0.747	0.752	0.527	0.551	0.558	0.564	0.488	0.552	0.598	0.625
DVSQ	**0.780**	**0.790**	**0.792**	**0.797**	**0.715**	**0.727**	**0.730**	**0.733**	**0.658**	**0.671**	**0.678**	**0.684**

Table 2. Mean Average Precision (MAP) Results of DVSQ and Its Variants, DVSQ-2 and DVSQ-C on Three Benchmark Datasets

Dataset	NUS-WIDE				CIFAR-10				ImageNet			
	8 bits	16 bits	24 bits	32 bits	8 bits	16 bits	24 bits	32 bits	8 bits	16 bits	24 bits	32 bits
DVSQ-C	0.654	0.663	0.665	0.676	0.603	0.611	0.625	0.634	0.448	0.477	0.504	0.521
DVSQ-2	0.752	0.763	0.773	0.782	0.672	0.685	0.693	0.705	0.634	0.648	0.662	0.671
DVSQ	**0.780**	**0.790**	**0.792**	**0.797**	**0.715**	**0.727**	**0.730**	**0.733**	**0.658**	**0.671**	**0.678**	**0.684**

by cross-validation from 10^{-5} to 10^{-1}. We adopt $K = 256$ codewords for each codebook as [8]. For each point, the binary code of all M codebooks requires $B = M \log_2 K = 8M$ bits (M bytes), where we set $M = B/8$ as B is known. We select parameters of all methods via cross-validation.

4.2. Results

The MAP results of all methods are listed in Table 1, which shows that the proposed DVSQ method substantially outperforms all the comparison methods. Specifically, compared to SQ, the best shallow quantization method with deep features as input, DVSQ achieves absolute increases of **10.0%**, **13.5%** and **12.2%** in average MAP on NUS-WIDE, CIFAR-10, and ImageNet respectively. Compared to DQN, the state-of-the-art deep quantization method, DVSQ outperforms DQN by very large margins of **5.1%**, **17.6%** and **10.7%** in average MAP on the three datasets respectively.

Compared to DQN, the advantages of DVSQ are four folds. (1) DVSQ can transfer semantic knowledge learned in general text domains (Google News in our case) to enable effective image retrieval by transforming all visual data to the semantic space, while DQN can only use the labeling information. (2) DVSQ can exploit the nonlinear correlation in the label space, while DQN assumes labels are independent. (3) DVSQ enables maximum inner-product search by exploiting both visual and semantic information, while DQN only supports minimum Euclidean distance search by exploiting the visual information. (4) From the formulation perspective, DVSQ adopts a well-designed adaptive margin loss for similarity-preserving learning from both image representations and label embeddings; DQN adopts a squared cosine loss to preserve the similarity between images, which cannot model true label distributions in the semantic space.

The retrieval performance in terms of Precision-Recall curves and Precision curves with respect to different numbers of top returned samples are shown in Figures 3 and 4, respectively. The proposed DVSQ model significantly outperforms all comparison methods by large margins in the two metrics. In particular, DVSQ achieves much higher precision at lower recall levels or when the number of top returned samples is small. This is desirable for precision-first retrieval, which is widely implemented in practical systems.

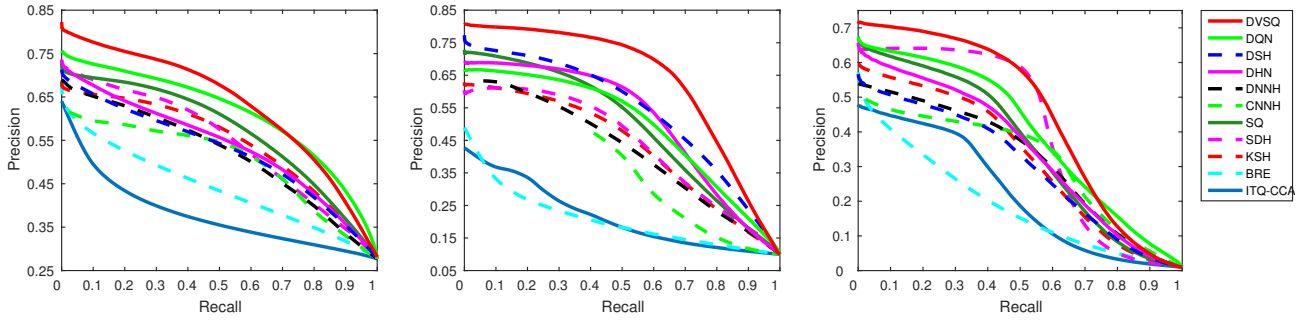
4.3. Discussion

We go deeper into the efficacy of DVSQ by three experiments: ablation study, visualization and sensitivity analysis.

4.3.1 Ablation Study

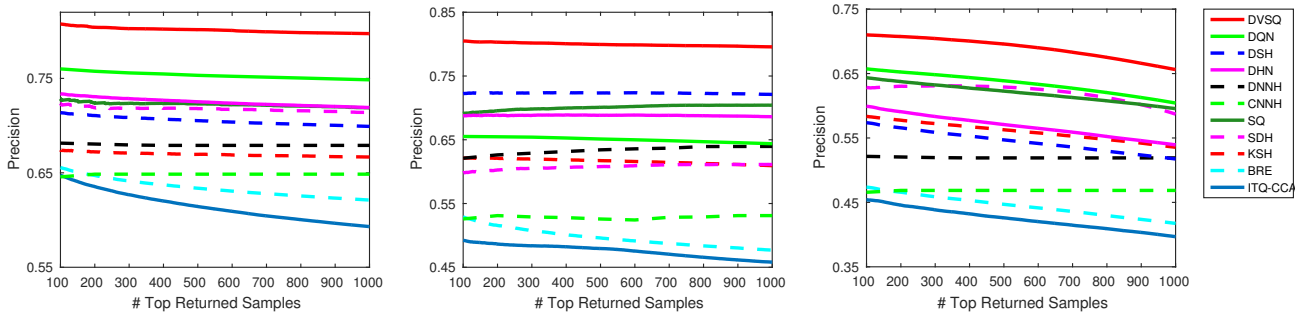
We investigate two DVSQ variants: (1) **DVSQ-2**, the two-step variant which separately learns deep visual-semantic embeddings via hybrid deep network (1) and generates binary codes via visual-semantic quantization (3); (2) **DVSQ-C**, a DVSQ variant that adopts softmax classifier to model the image labels instead of using the proposed adaptive margin loss, and generates binary codes from the deep features using CQ [44]. The MAP results w.r.t. different numbers of bits on three benchmark datasets are reported in Table 2.

Deep Visual-Semantic Embedding. Table 2 shows that DVSQ outperforms DVSQ-C by very large margins of 12.5%, 10.8% and 18.5% in average MAP on three datasets,



(a) NUS-WIDE (b) CIFAR-10 (c) ImageNet

Figure 3. Precision-recall curves on the NUS-WIDE, CIFAR-10 and ImageNet datasets with binary codes @ 32 bits.



(a) NUS-WIDE (b) CIFAR-10 (c) ImageNet

Figure 4. Precision@top-N curves on the NUS-WIDE, CIFAR-10 and ImageNet datasets with binary codes @ 32 bits.

respectively. DVSQ-C uses the standard softmax classifier which achieves state-of-the-art results on classification tasks, but this loss is not optimal for retrieval tasks. We thus propose an adaptive margin loss, which is an optimal rank loss that can guide deep visual-semantic embedding model to leverage the semantic knowledge in label embeddings.

Quantizability. Another observation is that by jointly preserving similarity from both image representations and label embeddings as well as controlling the quantization error of binary codes, DVSQ outperforms DVSQ-2 by 2.2%, 3.7% and 1.9% in average MAP on three datasets. This validates end-to-end quantization can improve the quantizability of deep representations and yield more accurate results.

4.3.2 Visualization

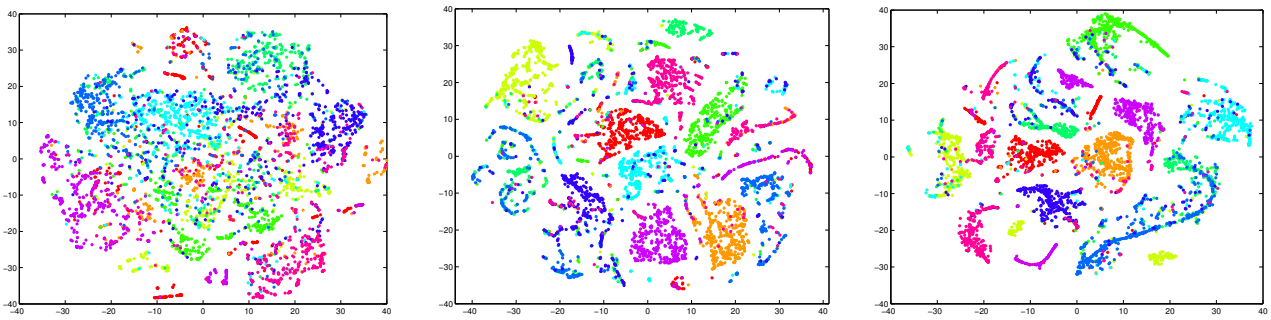
Figure 5 shows the t-SNE visualizations [36] of the deep representations learned by DQN [8], DVSQ-2, and DVSQ on CIFAR-10 dataset. As shown in Figure 5(a), DQN fails to capture the discriminative structures of the image data, due to its disability of leveraging the semantic information in label embeddings. DVSQ-2 addresses this issue by learning a deep visual-semantic embedding model to minimize the adaptive margin rank loss, hence it can learn more discriminative structures than DQN, as shown in Figure 5(b). DVSQ enables end-to-end learning of deep visual-semantic embeddings and compact binary codes, which enhances the quantizability of image embeddings, hence the image embeddings generated by DVSQ show most discriminative structures with clearest boundaries, as shown in Figure 5(c).

4.3.3 Sensitivity

Adaptive Margin. To evaluate the efficacy of the adaptive margin loss (2), we design an experiment that uses fixed-margin δ in the training procedure. We compute the MAP score @ 32 bits by varying the fixed-margin δ in $[0.1, 1.0]$, with results shown in Figure 6(a). We can observe that the optimal fixed-margin δ in different datasets are significantly different, e.g. $\delta = 0.9$ for NUS-WIDE and ImageNet, and $\delta = 0.6$ for CIFAR-10. The results reveal that different datasets may tolerate different fixed-margins which are determined by label distributions in the semantic space. This proves the importance of the proposed adaptive margin (2).

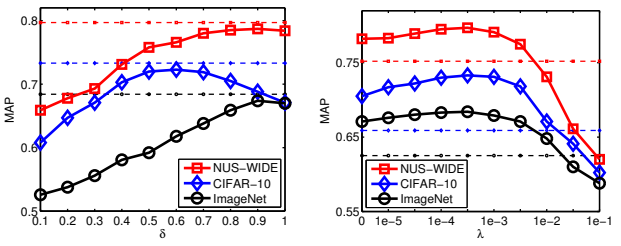
Also, DVSQ using optimal fixed-margin still underperforms DVSQ using adaptive margin (2) (dashed lines). Note that there is no need to specify a margin-parameter for the adaptive margin loss, which enables much easier model selection. These superior results validate that the proposed adaptive margin loss (1) is very effective for learning deep visual-semantic embeddings to establish accurate retrieval.

Parameter Sensitivity. We investigate the sensitivity of



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

Figure 5. t-SNE visualization of the deep representations of DQN [8], DVSQ-2, and DVSQ, with 32 bits codes on CIFAR-10 dataset.



(a) MAP w.r.t. δ @ 32 bits (b) MAP w.r.t. λ @ 32 bits

Figure 6. The MAP of DVSQ @ 32 bits w.r.t. fixed-margin $\delta \in [0.1, 1.0]$ and tradeoff parameter $\lambda \in [0, 0.1]$ on the three datasets.

λ, the parameter to trade off the importance of the quantization loss. We compute the MAP @ 32 bits by varying λ between 0 and 0.1. The MAP results of DVSQ with respect to different values of λ are shown in Figure 6(b). DVSQ consistently outperforms the best baseline method in a large range of $0 \leq \lambda \leq 0.01$. Note that when $\lambda \to 0$, DVSQ degenerates to its two-step variant DVSQ-2. The retrieval performance of DVSQ first increases and then decreases as λ varies and demonstrates a desirable bell-shaped curve. This justifies our motivation of jointly learning deep visual-semantic embeddings and visual-semantic quantization to improve the quantizability, since a good trade-off between them can enable learning of high-quality binary codes.

Encoding Time. In real applications, generating binary codes for new images should be fast. We compare encoding time of DVSQ with three deep methods DQN [8], DSH [25] and DHN [45] as well as two shallow methods SQ [39] and SDH [35]. We additionally consider shallow methods both using deep features (denoted by SQ and SDH) and without using deep features (denoted by SQ-D and SDH-D).

Figure 7 shows the encoding time (in logarithm) of all methods. DVSQ has comparable encoding time as deep methods DQN and DHN. Without using deep features, SQ-D and SDH-D take the least time, but after considering the time of deep feature extraction, SQ and SDH take almost the same time with DVSQ. Thus when using deep features,

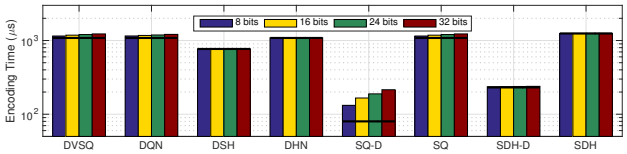


Figure 7. Encoding time of each image on NUS-WIDE dataset.

shallow methods are no longer superior to deep methods on the encoding time. The black lines on the bar chart split the times of feature extraction (below) and binary code generation (above). For hashing methods DSH, DHN, SDH and SDH-D, the binary codes can be generated by simple sign thresholding, which costs little time. For quantization methods DVSQ, DQN and SQ, generating binary code is slightly more costly than hashing methods, but is still an order of magnitude less than the deep feature extraction time.

5. Conclusion

This paper addressed deep learning to quantization from labeled image data and the semantic information extracted in general text domains. The proposed DVSQ model can learn compact binary codes by optimizing a novel adaptive margin loss and a visual-semantic quantization loss over the hybrid deep network. DVSQ can successfully transfer the semantic knowledge of labels learned from general text domains to enable more effective and efficient image retrieval. Comprehensive empirical evidence shows that DVSQ can generate compact binary encoding and yield state-of-the-art multimedia retrieval performance on standard benchmarks.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61502265, 61325008), National Key R&D Program of China (2016YFB1000701, 2015BAF32B01), National Engineering Lab for Big Data System Software (NEL-BDSS), and Tsinghua National Lab for Information Science and Technology (TNList) Projects.

References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, GA, 2016. USENIX Association.

[2] A. Babenko and V. Lempitsky. Additive quantization for extreme vector compression. In *CVPR*. IEEE, 2014.

[3] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(8):1798–1828, Aug 2013.

[4] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society*, 48(3):259–320, 1986.

[5] Y. Cao, M. Long, and J. Wang. Correlation hashing network for efficient cross-modal retrieval. *CoRR*, abs/1602.06697, 2016.

[6] Y. Cao, M. Long, J. Wang, and S. Liu. Collective deep quantization for efficient cross-modal retrieval. In *AAAI*. AAAI, 2017.

[7] Y. Cao, M. Long, J. Wang, Q. Yang, and P. S. Yu. Deep visual-semantic hashing for cross-modal retrieval. In *SIGKDD*, 2016.

[8] Y. Cao, M. Long, J. Wang, H. Zhu, and Q. Wen. Deep quantization network for efficient image retrieval. In *AAAI*. AAAI, 2016.

[9] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng. Nus-wide: A real-world web image database from national university of singapore. In *ICMR*. ACM, 2009.

[10] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014.

[11] C. Du and J. Wang. Inner product similarity search using compositional codes. *CoRR*, abs/1406.4966, 2014.

[12] V. Erin Liong, J. Lu, G. Wang, P. Moulin, and J. Zhou. Deep hashing for compact binary codes learning. In *CVPR*, pages 2475–2483. IEEE, 2015.

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

[14] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. A. Ranzato, and T. Mikolov. Devise: A deep visual-semantic embedding model. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2121–2129. Curran Associates, Inc., 2013.

[15] T. Ge, K. He, Q. Ke, and J. Sun. Optimized product quantization. *TPAMI*, 2014.

[16] A. Gionis, P. Indyk, R. Motwani, et al. Similarity search in high dimensions via hashing. In *VLDB*, volume 99, pages 518–529. ACM, 1999.

[17] Y. Gong, S. Kumar, H. Rowley, S. Lazebnik, et al. Learning binary codes for high-dimensional data using bilinear projections. In *CVPR*, pages 484–491. IEEE, 2013.

[18] Y. Gong and S. Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *CVPR*, pages 817–824, 2011.

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CVPR*, 2016.

[20] H. Jegou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(1):117–128, Jan 2011.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[22] B. Kulis and T. Darrell. Learning to hash with binary reconstructive embeddings. In *NIPS*, pages 1042–1050, 2009.

[23] H. Lai, Y. Pan, Y. Liu, and S. Yan. Simultaneous feature learning and hash coding with deep neural networks. In *CVPR*. IEEE, 2015.

[24] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2(1):1–19, Feb. 2006.

[25] H. Liu, R. Wang, S. Shan, and X. Chen. Deep supervised hashing for fast image retrieval. In *CVPR*, 2016.

[26] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang. Supervised hashing with kernels. In *CVPR*. IEEE, 2012.

[27] W. Liu, J. Wang, S. Kumar, and S.-F. Chang. Hashing with graphs. In *ICML*. ACM, 2011.

[28] X. Liu, J. He, B. Lang, and S.-F. Chang. Hash bit selection: a unified solution for selection problems in hashing. In *CVPR*, pages 1570–1577. IEEE, 2013.

[29] M. Long, Y. Cao, J. Wang, and P. S. Yu. Composite correlation quantization for efficient multimodal retrieval. In *SIGIR*, 2016.

[30] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[31] M. Norouzi and D. M. Blei. Minimal loss hashing for compact binary codes. In *ICML*, pages 353–360. ACM, 2011.

[32] M. Norouzi, D. M. Blei, and R. R. Salakhutdinov. Hamming distance metric learning. In *NIPS*, pages 1061–1069, 2012.

[33] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[34] R. Salakhutdinov and G. E. Hinton. Learning a nonlinear embedding by preserving class neighbourhood structure. In *AISTATS*, pages 412–419, 2007.

[35] F. Shen, C. Shen, W. Liu, and H. Tao Shen. Supervised discrete hashing. In *CVPR*. IEEE, June 2015.

[36] L. van der Maaten and G. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9: 2579–2605, Nov 2008.

[37] J. Wang, S. Kumar, and S.-F. Chang. Semi-supervised hashing for large-scale search. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(12):2393–2406, 2012.

[38] J. Wang, H. T. Shen, J. Song, and J. Ji. Hashing for similarity search: A survey. Arxiv, 2014.

[39] X. Wang, T. Zhang, G.-J. Qi, J. Tang, and J. Wang. Supervised quantization for similarity search. In *CVPR*, 2016.

[40] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *NIPS*, 2009.

[41] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan. Supervised hashing for image retrieval via image representation learning. In *AAAI*, pages 2156–2162. AAAI, 2014.

[42] F. X. Yu, S. Kumar, Y. Gong, and S.-F. Chang. Circulant binary embedding. In *ICML*, pages 353–360. ACM, 2014.

[43] P. Zhang, W. Zhang, W.-J. Li, and M. Guo. Supervised hashing with latent factor models. In *SIGIR*, pages 173–182. ACM, 2014.

[44] T. Zhang, C. Du, and J. Wang. Composite quantization for approximate nearest neighbor search. In *ICML*. ACM, 2014.

[45] H. Zhu, M. Long, J. Wang, and Y. Cao. Deep hashing network for efficient similarity retrieval. In *AAAI*. AAAI, 2016.