Cross-Modal Deep Variational Hashing

Venice Erin Liong[1,3], Jiwen Lu[2,*], Yap-Peng Tan[3], and Jie Zhou[2]
[1]Rapid-Rich Object Search (ROSE) Laboratory, Interdisciplinary Graduate School,
Nanyang Technological University, Singapore
[2]Department of Automation, Tsinghua University, Beijing, China
[3]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
veniceer001@e.ntu.edu.sg; lujiwen@tsinghua.edu.cn; eyptan@ntu.edu.sg; jzhou@tsinghua.edu.cn

Abstract

In this paper, we propose a cross-modal deep variational hashing (CMDVH) method for cross-modality multimedia retrieval. Unlike existing cross-modal hashing methods which learn a single pair of projections to map each example as a binary vector, we design a couple of deep neural network to learn non-linear transformations from image-text input pairs, so that unified binary codes can be obtained. We then design the modality-specific neural networks in a probabilistic manner where we model a latent variable as close as possible from the inferred binary codes, which is approximated by a posterior distribution regularized by a known prior. Experimental results on three benchmark datasets show the efficacy of the proposed approach.

1. Introduction

Recent years have witnessed that learning-based hashing is an active research topic for efficient large-scale multimedia search [8, 9, 16, 24, 29, 34]. The basic idea of learning-based hashing methods aims to learn a series of hash functions from the training set to map each visual sample into a compact binary feature vector such that samples of the same semantic content are mapped into same binary codes.

While recent works have achieved reasonably good performance in large-scale multimedia search, most existing hashing methods are developed for single-modal retrieval, which means that the query example and the examples stored in the database are from the same source of multimedia data. In many real-applications, it is easy to access multi-modal data for multimedia retrieval. For example, images uploaded into social networks such as the Flickr and Facebook websites are usually tagged with some text descriptions. Hence, it is desirable to retrieve semantically-similar texts/images by using a query image/text. Because there are large-scale multi-modal data over the Internet, it is only necessary to develop an effective cross-modal similarity search methods for multimedia search. In this paper, we propose a cross-modal deep variational hashing (CMDVH) method for cross-modality retrieval. Figure 1 illustrates the basic idea of the proposed approach. Unlike existing shallow cross-modal hashing methods which learn a single pair of linear or nonlinear projections to map each example into a binary vector, we employ an end-to-end hashing network to learn multiple pairs of hierarchical non-linear transformations, under which the nonlinear relationship of samples can be well exploited, the binarized neural codes having same semantic are similar as possible, and neural codes having different semantic are dissimilar as possible. Our model is trained under two main steps: First, we perform binary code inference to learn unified binary codes for each training pair using a cross-modal fusion network such that we obtain a common hamming space for the two modalities and the modality gap can be implicitly reduced. We perform this in a *discrete* and *discriminative* manner to avoid approximate optimization loss caused by relaxing the binary constraint and strengthen the semantic correlation between modalities by using a classification-based hinge loss criterion, respectively. Second, we model the modality-specific hashing networks which have a *probabilistic* interpretation such that the latent variable is modeled similar to the inferred binary code from the fusion network through a log likelihood criterion, which is also sampled based on an approximate posterior distribution regularized by a prior through a Kullback-Liebler Divergence (KLD) criterion. By doing so, the hashing network can be in generative form, which is suitable for out-of-sample extension. We perform learning in these two steps through a batch-wise gradient descent procedure. Experimental results on three benchmark datasets show the efficacy of the proposed approach.

[*] Corresponding author.

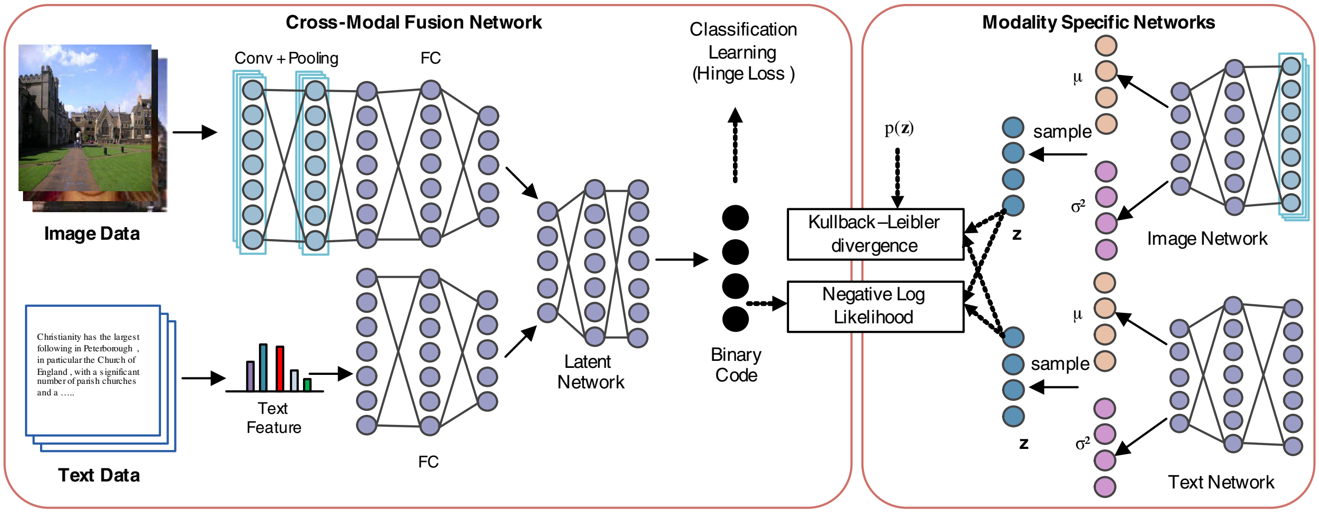

Cross-Modal Fusion Network
Conv + Pooling
FC
Image Data
Christianity has the largest following in Peterborough , in particular the Church of England , with a significant number of parish churches and a …..
Text Feature
Text Data
FC
Latent Network
Classification Learning (Hinge Loss)
Binary Code
p(z)
Kullback–Leibler divergence
Negative Log Likelihood
Modality Specific Networks
μ
sample
σ²
z
Image Network
μ
sample
z
σ²
Text Network

Figure 1. The basic idea of our proposed approach for cross-modality multimedia retrieval. Given a gallery set represented by two modalities (image and text), we learn a fusion hashing network and a joint binary code matrix, simultaneously. We learn them using an alternative optimization procedure. First, we infer the binary codes in discrete manner such that we exploit label information through a classification based hinge-loss criterion. Second, we minimize the loss between neural code and binary code by performing end-to-end deep training via backpropagation to learn the parameters of the each network. This is done iteratively until convergence. Once the inferred binary codes are learned, we learn modality-specific hashing networks (one for each modality) such that a latent variable is modeled based on two criterions. First, given the image-text pair, the latent variable is forced to be similar as possible to the inferred binary code from the fusion network through a negative log likelihood criterion. Second, the latent variable is also modeled such that approximated posterior distribution in the form of Multivariate Gaussian is close to prior regularized by the KL-divergence criterion. During retrieval, given a query, we extract the query binary code using the learned modality-specific hashing network and obtain the most similar binary codes from the gallery (learned $\mathbf{B}$) which are indexed to retrieve the most relevant images.

2. Related Work

Cross-Modal Retrieval: Unlike single-modal retrieval where both the query example and the database are from the same modality, the key idea of cross-modal retrieval is to retrieve samples from another modality which is different from that of the query example but share similar semantics. Typically, cross-modal multimedia retrieval perform two main tasks: 1) retrieval of text documents by using a given query image, and 2) retrieval of images by using a given query text or tag. In recent years, several methods have been proposed for cross-modal retrieval, where the objective is to learn a common subspace between images and text [22, 31] to model the correlations. For example, Rasiwasia *et al.* [22] used canonical component analysis (CCA) to map both text documents and images into a latent space. Wang *et al.* [31] learned a coupled feature space method to select the most relevant and discriminative features for cross-modal matching. Gong *et. al.* [7] performed nonlinear kernel embedding followed by a linear dimensionality reduction and CCA for content-based retrieval and tag-image search. Kang *et. al.* [11] proposed a feature learning approach for cross-media matching by jointly learning consistent features for each modality in a supervised manner. More recently, Wang *et al.* [30] employed a feature selection scheme using multimodal-graph to represent the similarity between modalities. These retrieval methods usually perform cross-modal matching with high-dimensional features, hence are not suitable for large-scale search due to the scalability issue. Therefore, hashing is a more desirable choice for cross-modal retrieval.

Shallow Cross-Modal Hashing: In recent years, several cross-modal hashing methods have also been proposed in the literature, and most studies are in shallow form in which it only performs a single-layer of linear or nonlinear transformation. These can be classified into two types: *unsupervised* [5, 28, 37] and *supervised* [1, 17, 35, 36]. Unsupervised methods utilize co-occurence information such that only the image-text pairs which occured in the same article are known to be of similar semantic. For example, Kumar *et al.* [14] presented a cross-modal spectral hashing method so that the cross-modality similarity is also preserved in the learned hash functions. Zhu *et al.* [38] learned a common latent space by preserving the similarity between the example to the k-nearest centroids in each modality and cross-modality. Zhou *et al.* [37] obtained a unified binary from a latent space learning method by using sparse coding and matrix factorization in the common space. Ding *et.al.* [5] learned a unified binary code in the training stage by performing matrix factorization with latent factor model. Supervised methods utilize semantic labels to enhance the correlation of cross-modal data. For example, Brostein *et al.* [1] presented a cross-modal hashing method by preserving the intra-class similarity through eigen-decomposition and boosting. Zhang *et al.* [36] performed semantic correlation maximization using label information to learn a modality-specific transformations which maximizes the correlation between modalities. Lin *et.al.* [17] learned a unified binary code by modeling them in a probability distribution in a supervised manner and performed kernel-embedding to learn the hashing functions. Xu *et.al.* [35] also learned a unified binary code and used a linear classifier to exploit the label information.

Unlike these methods which learn a pair of linear/nonlinear projections for hash functions learning, we employ hashing networks to learn multiple pairs of hierarchical non-linear transformations, so that the nonlinear relationship of samples and the relationship of samples from different modalities can be well exploited. Cross-modal hashing methods can also be classified as learning a joint binary code or separate binary codes during training. Several recent works learned unified binary codes [5,17,35,37] and these methods generally showed better performance because by learning a single discriminative and efficient binary code, the modality gap between the hashing functions are implicitly reduced. Hence, we also perform a shared binary code learning strategy in our hash function learning procedure, then perform modality-specific hash function learning to have a generative model.

Deep Cross-Modal Hashing: Over the past few years, a variety of deep learning algorithms have been proposed in machine learning, and some of them were successfully applied to many computer vision applications such as in object detection and recognition [13, 25]. While there are now also studies that perform deep learning for cross-modal retrieval [21,27,32,33] they are not suitable for large-scale search due to its high dimension and large storage requirement. Only few works have performed deep learning for cross-modal hashing. For example, Masci *et al.* [19] learned a similarity preserving network for cross-modalities through a coupled siamese network with hinge loss. However this does not consider the binary constraints during training, and simply performs binarization after training. Cao *et al.* [3] designed a stacked auto-encoder architecture to jointly maximize the feature and semantic correlation across modalities. However, this work does not perform end-to-end learning which may limit the discriminative representation of data samples, particularly in images. Jiang *et al.* [10] performed an end-to-end deep learning framework with a negative log likelihood criterion to preserve the similarity between real-value representations having the same class. However, their training model performs similarity preservation on real-value codes and not binary codes which are used for the actual retrieval during testing. Another work from Cao *et al.* [2] learned a visual semantic fusion network with cosine hinge loss, to obtain the binary codes and learned modality-specific deep networks to obtain the hashing functions. However, a metric-based approach may not fully utilize the label information during training.

3. Cross-Modal Deep Variational Hashing

We propose an end-to-end deep architecture for cross-modal hashing such that we are able to implicitly maximize the correlation between the two modalities given image-text training data pairs and its corresponding label information. Our implementation composes of a fusion network for binary code inference that learns binary codes from image and text data discretely and discriminatively, and a generative modality-specific network to encode the image/text sample to representative binary codes. We now present these networks and how to perform optimization in the proceeding subsections.

Cross-Modal Fusion Network: Let $\mathbf{X}_u = [\mathbf{x}_{u1}, \mathbf{x}_{u2}, \cdots, \mathbf{x}_{uN}] \in \mathbb{R}^{d_u \times N}$ and $\mathbf{X}_v = [\mathbf{x}_{v1}, \mathbf{x}_{v2}, \cdots, \mathbf{x}_{vN}] \in \mathbb{R}^{d_v \times N}$ be the training sets from different modalities, where u and v represent two different modalities, N is the number of training samples in each modality, and $\mathbb{R}^{d_u}$ and $\mathbb{R}^{d_v}$ are the feature dimension for each sample in modalities u and v, respectively. Our fusion network aims to transform the cross-modal sample pair into a compact binary feature vector as follows:

$$f_{u,v} : (\mathbb{R}^{d_u}, \mathbb{R}^{d_v}) \rightarrow \{-1, 1\}^K \tag{1}$$

where K is the length of the binary feature vector. Specifically, for image and text as the modality pairs, the fusion network would comprise of convolution, pooling layers and FC layers with parameters θ_u to process the images, and FC layers with parameter θ_v to process the text data. To combine the output of two networks, we create a latent network which composes of FC layers with parameters θ_w. The input and output of the latent layer would be as follows:

$$\begin{aligned} w &= s(f_u(\mathbf{X}_u, \theta_u) + f_v(\mathbf{X}_v, \theta_v)) && (2) \\ \mathbf{h} &= f_w(w, \theta_w) && (3) \end{aligned}$$

where f_u, f_v and f_w are the image, text and latent network functions, respectively, and $s(\cdot)$ is the non-linear activation function. The output of the fusion network would then be $\mathbf{h} \in \mathbb{R}^{1 \times K}$. We let the output for the whole training set of the fusion network be $\mathbf{H} \in \mathbb{R}^{N \times K}$, the learned binary code matrix be $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_N] \in \{-1, 1\}^{N \times K}$, the label data be defined as $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N] \in \{1, 0\}^{N \times C}$ where $\mathbf{y}_{n,j} = 1$ if the n-th sample belongs to class j and 0 otherwise, and a multi-class projection matrix be defined as $\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_C] \in \mathbb{R}^{K \times C}$. We learn the binary code and network parameters in a discrete manner such that we preserve the binary property and avoid the approximation loss caused by relaxation, but also learn discriminative binary codes that are semantically correlated. This can be done by the following optimization procedure:

$$\begin{aligned}\min_{\mathbf{B},\mathbf{M},\theta_u,\theta_v,\theta_w} J &= J_1 + \lambda J_2 \\ &= \|\mathbf{M}\|_F^2 + \sum_n^N \xi_n + \lambda(\|\mathbf{B}-\mathbf{H}\|_F^2) \\ & \forall n,j \;\; \mathbf{y}_{n,j}(\mathbf{m}_j^\top \mathbf{b}_n) \geq 1 - \xi_n \\ & \forall n \;\; \mathbf{b}_n = \{-1,1\} \qquad (4)\end{aligned}$$

where J_1 minimizes the multi-classification loss formed from the hinge loss between the label information and binary code so that samples that are semantically relevant(irrelevant) have similar(dissimilar) binary codes as much as possible. J_2 minimizes quantization loss between the real-value code and binary code such that the energy of the samples can be well-preserved in the hashing network. Here, $\xi_n \geq 0$ is the slack variable and λ is a constant parameter to balance the effect of the two parameters.

The optimization problem in (4) is non-convex due to the binary constraints, which makes it difficult to solve. However, it can be addressed using an iterative approach where we keep other variables fixed and solve one alternatively and iteratively. We learn the binary code, multi-class projection matrix and network parameters $\theta = \{\theta_u, \theta_v, \theta_w\}$ as follows:

Update $\mathbf{M}$ *with fixed* $\mathbf{B}$ *and* θ*:* We are left with a support vector machine (SVM) formulation which can be solved through a standard solver[1] to learn the classification matrix $\mathbf{M}$.

Update $\mathbf{B}$ *with fixed* $\mathbf{M}$ *and* θ*:* We perform a discrete optimization technique and simplify (4) as follows to learn $\mathbf{B}$:

$$\begin{aligned}\min_{\mathbf{b}_n} J(\mathbf{b}_n) &= -\sum_{j=1}^{C} \mathbf{m}_{c_n,j}^\top \mathbf{b}_n \\ &+ \lambda\|\mathbf{b}_n - \mathbf{h}_n\|_F^2 \\ \text{subject to} \quad & \mathbf{b}_n \in \{-1,1\}^{1\times K} \qquad (5)\end{aligned}$$

(5) is a binary quadratic problem that can be solved through a linear gradient technique similar to [18]. We obtain a closed-form solution as follows:

$$\mathbf{b}_n = \operatorname{sgn}(\mathbf{y}_n \mathbf{M}^\top + \lambda \mathbf{h}_n) \qquad (6)$$

Update θ *with fixed* $\mathbf{M}$ *and* $\mathbf{B}$*:* We obtain the resulting formulation:

$$\min_{\theta} J(\theta) = \lambda\|\mathbf{B}-\mathbf{H}\|_F^2 \qquad (7)$$

```
Algorithm 1: CMDVH - cross-modal fusion network
Input: Training set $\mathbf{X}_u$ and $\mathbf{X}_v$, network learning
       parameters, iterative number $Iter$, objective
       function parameter $\lambda$ and convergence error $\epsilon$.
Output: unified binary code matrix $\mathbf{B}$
Step 1 (Initialization):
1.1 Initialize image, text and latent network parameters (see
Implementation details)
1.2 Initialize binary code $\mathbf{B}$, randomly and zero-centered.
Step 2 (Fusion Network and Binary Code Learning):
for $t = 1, 2, \cdots, Iter$ do
    - Compute $\mathbf{H}$ using the initial fusion network.
    2.1 (Classification Step):
    - Obtain $\mathbf{M}$ by solving the SVM formulation in (4).
    2.2 (Binary Code Learning Step):
    - Obtain $\mathbf{B}$ according to (6).
    2.3 (Hash Function Learning Step):
    - Obtain the top-layer gradients according to (8).
    - Perform back propagation for the image, text and
    latent network.
    - Calculate $J_t$ using (4).
    If $t > 1$ and $|J_t - J_{t-1}| < \varepsilon$.
end
Return: $\mathbf{B}$.
```

We employ the batch-wise gradient descent method to learn parameters for the latent network and image/text networks. The gradient of J in (7) with respect to the neural code representation are as follows:

$$\frac{\partial J}{\partial \mathbf{H}} = -2\lambda(\mathbf{B}-\mathbf{H}) \qquad (8)$$

For each layer of the network, the gradients can easily be computed through the chain rule during backpropagation. The parameters of the networks are updated using these gradients based on a given learning rate, momentum and weight decay. **Algorithm 1** summarizes the detailed procedure of our the cross-modal fusion network of our CMDVH.

Modality-Specific Networks: After learning a representative binary code for the training cross-modal pairs from a fusion network, we can now learn generative modality-specific networks for encoding out-of-sample input. The aim of modality-specific networks is to directly map each cross-modal sample pair into similar binary code inferred from the fusion network as follows :

$$g_u : \mathbb{R}^{d_u} \to \{-1,1\}^K, \quad g_v : \mathbb{R}^{d_v} \to \{-1,1\}^K \qquad (9)$$

Inspired by the success of variational encoders [12], we employ a probabilistic interpretation for the modality-specific network to make it more general and suitable for out-of-sample extension. We assume that the output data is generated by a latent variable, $\mathbf{z}$, sampled from a conditional distribution. Given data $\mathbf{x}_{*i}$[2], we assume that the latent

[1] we use LibSVM: http://www.csie.ntu.edu.tw/ cjlin/libsvm/

[2] where $* = \{u,v\}$.

sample and binary code is generated as $\mathbf{z}_{*i} \sim p_{\theta_*}(\mathbf{z}_{*i})$ and $\mathbf{b}_i \sim p_{\theta_*}(\mathbf{b}_i|\mathbf{z}_{*i})$, respectively. Similar to [12], we generate a proposal distribution $q_{\phi_*}(\mathbf{z}_*|\mathbf{x}_{*i})$ to approximate the posterior distribution $p_{\theta_*}(\mathbf{z}_{*i}|\mathbf{x}_{*i})$ where we sample $\mathbf{z}_{*i}$ as follows:

$$\begin{aligned} \mathbf{z}_{*i}^l &= \mu_{*i} + \sigma_{*i} \odot \epsilon^l \\ & \quad \epsilon^l \sim \mathcal{N}(0,1) \end{aligned} \tag{10}$$

where ϵ^l means the l-th sample of noise, $\odot$ denotes element-wise multiplication, μ_{*i} and σ_{*i} would be the output of the non-linear projection from network $g(\mathbf{x}_{*i}, \theta_*)$ with input $\mathbf{x}_{*i}$ and parameter θ_*. From (9), we can have the proposal distribution to be:

$$q_{\phi_*}(\mathbf{z}_{*i}|\mathbf{x}_{*i}) = \mathcal{N}(\mathbf{z}_{*i}|\mu_{*i}, \sigma_{*i}^2\mathbf{I}) \tag{11}$$

We also assume that the prior over the latent variable is centered by a multivariate gaussian $p_{\theta_*}(\mathbf{z}_*) = \mathcal{N}(\mathbf{z}_*; 0, \mathbf{I})$. From this assumption, we can derive the analytic form of the Kullback-Liebler (KL) divergence as:

$$\begin{aligned} D_{KL}(q_{\phi_*}(\mathbf{z}_{*i}|\mathbf{x}_{*i})||p_{\theta_*}(\mathbf{z}_{*i})) &= \frac{1}{2}\sum_{j=1}^{J}(1 + \log((\sigma_{*i}^{(j)})^2 \\ &- (\mu_{*i}^{(j)})^2 - (\sigma_{*i}^{(j)})^2) \end{aligned} \tag{12}$$

where j is the j-th element of μ and σ. The KL divergence would act as a regularizer to the approximate posterior distribution. Finally, In order to ensure that the latent variable produces binary codes similar to the learned codes in the fusion network, we employ a probabilistic loss function in the form of a log-likelihood loss as follows:

$$\log p(b_i^{(k)}|z_{*i}^{(k)}) = \log(1 + e^{b_i^{(k)} z_{*i}^{(k)}}) \tag{13}$$

where k is the k-th bit of the binary code. From these approximations, the network learning formulation can then be written as follows:

$$\begin{aligned} \min_{\theta} \mathcal{L} &= \sum_{i=1}^{N}\sum_{k=1}^{K}\mathcal{L}_{NLL} + \sum_{i=1}^{N}\alpha\mathcal{L}_{KLD} \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K} -\log(1 + e^{b_i^{(k)} z_{*i}^{(k)}}) \\ &- \frac{\alpha}{2}\sum_{i=1}^{N}\sum_{j=1}^{J}(1 + \log((\sigma_{*i}^{(j)})^2 - (\mu_{*i}^{(j)})^2 - (\sigma_{*i}^{(j)})^2) \end{aligned} \tag{14}$$

$\mathcal{L}_{NLL}$ ensures that the binary data likelihood under the approximate posterior distribution is maximized. $\mathcal{L}_{KLD}$ ensures that the KL divergence between the proposed distribution and prior distribution for the latent variable is minimized. Finally, α is a constant parameter to balance the two loss terms. (14) can be easily optimized by taking the gradient of the objective function and performing batch-wise backpropagation. **Algorithm 2** summarizes the detailed procedure of the modality-specific networks of our CMDVH.

```
Algorithm 2: CMDVH - modality-specific network
Input: Training set $\mathbf{X}_u$ and $\mathbf{X}_v$ with corresponding
       binary code matrix $\mathbf{B}$, network learning
       parameters, iterative number $Iter$, objective
       function parameter $\alpha$, and convergence error $\epsilon$.
Output: Network parameters $\theta_u$ and $\theta_v$
Step 1 (Initialization):
1.1 Initialize modality-specific network parameters (see
Implementation details)
Step 2 (Modality-Specific Hashing Network Learning):
for $*$ = image ($u$), text ($v$) do
    for $t = 1, 2, \cdots, Iter$ do
        2.1 (Forward Propagation):
        - Compute output of modality-specific network,
        given input sample $\mathbf{x}_*$.
        - Split output to $\mu_*$ and $\sigma_*$.
        - Sample $\mathbf{z}_*$ from (10).
        2.2 (Backward Propagation):
        - Compute gradient of loss function (14).
        - Perform gradient descent to learn $\theta_*$
    end
    Calculate $\mathcal{L}_t$ using (14).
    If $t > 1$ and $|\mathcal{L}_t - \mathcal{L}_{t-1}| < \varepsilon$.
end
Return: $\{\theta_u, \theta_v\}$.
```

For new instances or query data, we simply use the learned modality-specific networks to obtain the output real-value codes and finally binarize them using the sign($\cdot$) function. During retrieval, given a text query (can be image), we extract the query binary code using the learned text hashing network and obtain the most similar binary codes from the gallery (learned $\mathbf{B}$) which are indexed to retrieve the most relevant images.

4. Experiments

We conducted experiments on three widely used datasets to evaluate our CMDVH. The following describes the details of the experiments and results.

4.1. Datasets and Experimental Setup

Datasets: We employed three cross-modal datasets in our experiments: Wiki, IAPRTC12 and NUS-WIDE. The Wiki dataset[3] contains 2866 Wikipedia documents, where each document contains a single image and a corresponding text of at least 70 words. These documents are categorized

[3] http://www.svcl.ucsd.edu/projects/crossmodal/.

into 10 semantic classes, where each document is from one class. Each text is represented by a 10-dimensional feature vector which is computed from the latent Dirichlet Allocation (LDA) model. We randomly selected 75% documents from this dataset as the database and the rest as query samples.

The IAPR TC-12 dataset[4] contains 19627 images with corresponding sentence descriptions. These image-sentence pairs present various semantics such as landscape, action and people categories. Similar to [2], we use the top 22 frequent labels from the 275 concepts obtained generated from the segmentation task[5]. For the text features, we pre-process the sentence data removing the stop words and extract a bag-of-words (BoW) representation with a dimension of 500. We randomly select 100 pairs per class as the query set and the remaining data as the gallery set. Unlike the Wiki where each image was associated with one category class, the images in IAPRTC12 may have more than one label information.

The NUS-Wide dataset[6] contains 269648 images which were annotated by 81 concept tags. Following the same settings in previous works [14, 23], we selected the 10 most frequent concepts and constructed a subset which contains 186577 images-tag pairs. Similar to the IAPRTC12, each image in the NUS-WIDE dataset is associated with multiple tags. In our experiments, each text is represented by a 1000-dimensional feature vector which is computed by the bag-of-words model. We randomly selected 99% samples to form the database and the rest as query samples.

Evaluation Metrics: For each dataset, we performed two cross-modal retrieval tasks: image-to-text retrieval and text-to-image retrieval, which search texts by a query image and search images by a query text, respectively. We use the mean average precision (mAP) [1,14,23] to measure the performance of different retrieval methods, which is defined as the mean of all queries' average precision, AP, defined as follows:

$$AP = \frac{1}{M}\sum_{r=1}^{R} prec(r) \odot rel(r) \tag{15}$$

where M is the number of relevant instances in the retrieved set, $prec(r)$ denotes the precision of the top r retrieved set, and $rel(r)$ is an indicator of relevance of a given rank (which is set to 1 if relevant and 0 otherwise). Here, we consider two samples similar as long as there is at least one similar label. In our experiments, we use $R = 100$ for the NUS-WIDE and Wiki dataset, and $R = 500$ for the IAPRTC12. Generally, mAP measures the discriminative learning ability of different cross-modal retrieval methods, where a higher mAP indicates better retrieval performance.

Because the IAPRTC12 and NUS-WIDE dataset have multiple labels for each sample, it is important that a ranking metric is also evaluated. Hence, we also evaluate the Normalized Discounted Cumulative Gain (NDCG), and Average Cumulative Gain (ACG). For a given query sample $\mathbf{x}_q$, these criterions are defined as follows:

$$NDCG@p = \frac{1}{Z}\sum_{i=1}^{p}\frac{2^{r_i}-1}{\log(1+i)} \tag{16}$$

$$ACG@p = \frac{1}{p}\sum_{i=1}^{p} r_i \tag{17}$$

where Z is the normalized constant, r_i is the similarity level of the ith sample, and p is the number of retrieved samples in the ranking list. r_i represents a ranking level valued z is the query and i-th sample in gallery share z similar labels, and valued zero if they do not share any label. The NDCG evaluates the ranking by penalizing errors in higher ranked items more strongly, while ACG takes the average of the similarity levels of data within the retrieved samples.

Implementation Details: Our deep architecture and experiments were implemented under the MatConvNet [26] framework. For the fusion network, the *image hashing network* used the pre-trained CNN-F from [4] as our initial convolution and pooling layers up to FC7, and stack a number of new FC layers with dimensions of [4096 $\rightarrow$ 500 $\rightarrow$ 200] for all datasets, while the *text hashing network* is designed with fully-connected networks and use the pre-processed text features, given by each experiment, as input. We set the FC layers as [10 $\rightarrow$ 100 $\rightarrow$ 200], [1386 $\rightarrow$ 500 $\rightarrow$ 200], and [1000 $\rightarrow$ 500 $\rightarrow$ 200], for the Wiki, IAPRTC12, and NUS-WIDE dataset, respectively. For the latent network which fuses the output of image and text network, we used FC layers with dimensions of [200 $\rightarrow$ 500 $\rightarrow$ K]. For the modality-specific networks, we use the similar image and text networks except that a the top FC layer would have a size of $2 \times K$ because of the splitting done during latent variable sampling. We perform end-to-end learning by having the learning rate at the new fully connected layers to be 0.01. To avoid overfitting and ruining the representative abstract features already learned during the pre-training, we reduce the learning rate of the remaining convolution and FC layers to be 0.0001. For both image and text network, we used the ReLU activation[7] as the nonlinear activation function for the new fully connected layers except for the last layer. We use the hyperbolic tangent (tanh) function for the top layer of the latent network because it is able to squeeze the representation to a {-1,1} range which ensures that the quantization loss can be reduced as much as possible. The parameters in the new fully connected layers are initialized using the Xavier ini-

[4] http://imageclef/photodata.

[5] http://imageclef/SIAPRdata.

[6] http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm

[7] ReLU is a nonlinear transformation f(x) = max(0, x) [20]

Table 1. mAP performance of different cross-modal hashing methods on different datasets, where images were used as query samples and texts/tags were employed as gallery samples, respectively.

	Wiki				IAPRTC12				NUS-WIDE			
Method	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
CVH [14]	0.2383	0.2038	0.1791	0.1580	0.5370	0.5409	0.5242	0.4962	0.5045	0.5484	0.5588	0.5583
CCA-ITQ [8]	0.3328	0.3216	0.3064	0.328	0.5587	0.5853	0.5895	0.5855	0.5400	0.5960	0.6194	0.6229
PDH [23]	0.3251	0.3258	0.3436	0.3438	0.5927	0.6085	0.6302	0.6450	0.5687	0.6148	0.6475	0.6793
LSSH [37]	0.3645	0.3713	0.3777	0.3580	0.5440	0.5769	0.5964	0.5985	0.5547	0.5734	0.5980	0.5968
CMFH [5]	0.2665	0.2755	0.2876	0.2950	0.5601	0.5829	0.6079	0.6179	0.4772	0.5301	0.5763	0.6258
SCM [36]	0.1387	0.1367	0.1413	0.1359	0.5665	0.5051	0.4548	0.4178	0.5190	0.4837	0.4495	0.4189
SePH - *km* [17]	0.4144	0.4354	0.4374	0.4472	0.6177	0.6447	0.6500	0.6781	0.6524	0.6526	0.6637	0.6696
DisCMH [35]	0.3754	0.3936	0.3901	0.3915	0.6174	0.6596	0.6503	0.6594	0.6826	0.7583	0.7752	0.7605
CMDVH	**0.4242**	**0.4430**	**0.4519**	**0.4442**	**0.7196**	**0.7727**	**0.8004**	**0.7902**	**0.8503**	**0.8755**	**0.8801**	**0.8910**

Table 2. mAP performance of different cross-modal hashing methods on different datasets, where texts/tags were used as query samples and images were employed as gallery samples, respectively.

	Wiki				IAPRTC12				NUS-WIDE			
Method	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
CVH [14]	0.3882	0.3362	0.2567	0.2297	0.5677	0.5784	0.5610	0.5362	0.5280	0.5732	0.5864	0.5807
CCA-ITQ [8]	0.5463	0.5505	0.5593	0.5633	0.5863	0.6123	0.6143	0.6053	0.5753	0.6151	0.6405	0.6360
PDH [23]	0.5432	0.5592	0.57554	0.58474	0.5960	0.6133	0.6345	0.6488	0.5844	0.6402	0.6817	0.7087
LSSH [37]	0.6061	0.6256	0.6384	0.6376	0.4868	0.5264	0.5547	0.5724	0.5857	0.6242	0.6293	0.6464
CMFH [5]	0.3955	0.4105	0.4473	0.4807	0.5592	0.5834	0.6084	0.6187	0.4965	0.5432	0.5995	0.6405
SCM [36]	0.1322	0.1429	0.1556	0.1494	0.6521	0.5697	0.4776	0.4213	0.5485	0.5033	0.4481	0.3920
SePH - *km* [17]	0.7007	0.6999	0.7099	0.7153	0.6105	0.6340	0.6404	0.6730	0.6604	0.6766	0.7043	0.7024
DisCMH [35]	0.6772	0.6602	0.6632	0.6537	0.6532	0.6910	0.6921	0.6949	0.6519	0.7378	0.7535	0.7511
CMDVH	**0.7270**	**0.7326**	**0.7383**	**0.7371**	**0.7348**	**0.7744**	**0.8038**	**0.8111**	**0.8270**	**0.8328**	**0.8403**	**0.8782**

Table 3. mAP performance of different deep cross-modal hashing methods on different datasets.

IAPRTC12					
	Method	16 bits	32 bits	64 bits	128 bits
$I \rightarrow T$	DNH-C [15]	0.5250	0.5592	0.5902	0.6339
	DVSH [2]	0.5696	0.6321	0.6964	0.7236
	CMDVH	**0.7196**	**0.7727**	**0.8004**	**0.7902**
$T \rightarrow I$	DNH-C [15]	0.4692	0.4838	0.4905	0.5053
	DVSH [2]	0.6037	0.6395	0.6806	0.6751
	CMDVH	**0.7348**	**0.7744**	**0.8038**	**0.8111**
NUSWIDE					
	Method	16 bits	32 bits	64 bits	128 bits
$I \rightarrow T$	CAH [3]	0.4920	0.5084	0.5407	0.5628
	DCMH [10]	0.6249	0.6355	0.6720	-
	CMDVH	**0.8503**	**0.8755**	**0.8801**	**0.8910**
$T \rightarrow I$	CAH [3]	0.5019	0.5135	0.5451	0.5800
	DCMH [10]	0.6791	0.6829	0.6906	-
	CMDVH	**0.8270**	**0.8328**	**0.8403**	**0.8782**

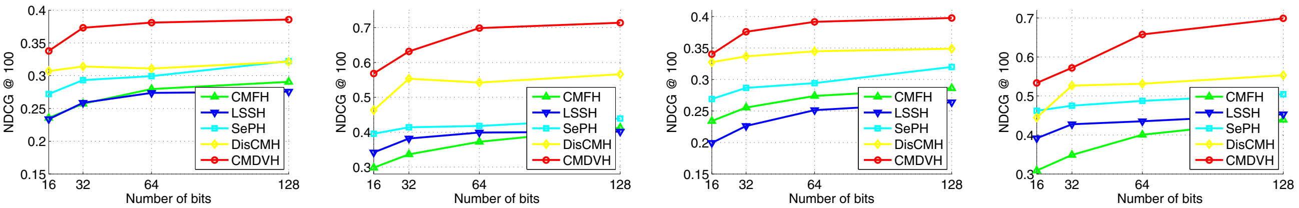
tialization [6][8]. The momentum, and weight decay were set to 0.9, and 0.0001, respectively. In our experiments, the parameters λ_1 and α were set to 0.2 and 0.5, respectively, which were obtained by cross-validation on the Wiki dataset using 16 bits.

4.2. Experimental Results

Comparisons with State-of-the-art Cross-Modal Hashing Methods: We compared our CMDVH with the different state-of-the-art cross-modal hashing methods which can be grouped to unsupervised (CVH, PDH, CCA-ITQ, LSSH, CMFH) and supervised (SCM, SePH, DisCMH).[9] To have a fair comparison because they are shallow methods, we make use of CNN features extracted at the FC7 layer for the images from the pre-trained model initially used by our CMDH method. Also, to maximize the learning potential of each dataset, we made use of the gallery samples as training data to learn the hashing functions. During retrieval, methods that employ unified binary code learning (LSSH, CMFH, SePH, DisCMH) similar to CMDH use the learned binary code as gallery set, while other methods (CVH, PDH, CCA-ITQ, SCM) use the learned hash function to obtain the binary codes for the gallery set. Tables 1 and 2 show the mAP performance by Hamming Ranking. It can be observed that our method provided the best performance compared to the shallow cross-modal hashing methods. This may be because our DCNN model captured the nonlinearities of the raw data due to several nonlinear transformations. Although SePH also performed nonlinear transformations, it was done explicitly through kernels which cannot really maximize the information from raw data. The DisCMH method gave competitive results with our CMHN method at lower bits, but did not consistently improve as the bit size increased. This may be because it performed linear projection which may have limited the binary code mapping. In addition, a larger performance gap can be seen in the IARPTC12 and NUSWIDE experiments most probably due to larger

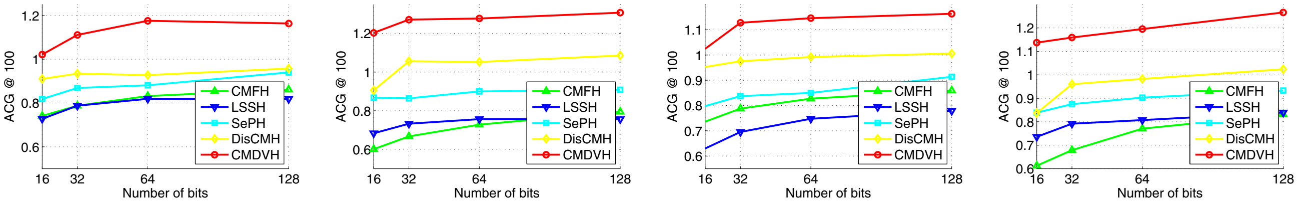
[8] $\mathbf{W} = U\left[-\sqrt{\frac{6}{n_{in}+n_{out}}}, \sqrt{\frac{6}{n_{in}+n_{out}}}\right]$ where $\mathbf{W} \in \mathbb{R}^{n_{in} \times n_{out}}$

[9] Authors provided their codes except for DisCMH in which we implemented ourselves.



(a) IAPRTC12 ($I \to T$) (b) IAPRTC12 ($T \to I$) (c) NUSWIDE ($I \to T$) (d) NUSWIDE ($T \to I$)

Figure 2. NDCG performance of different cross-modal hashing methods for the IAPRTC12 and NUSWIDE database.



(a) IAPRTC12 ($I \to T$) (b) IAPRTC12 ($T \to I$) (c) NUSWIDE ($I \to T$) (d) NUSWIDE ($T \to I$)

Figure 3. ACG performance of different cross-modal hashing methods for the IAPRTC12 and NUSWIDE database.

training data which hashing network training fully utilized. Figures 2- 3 show the NDCG and ACG performance. Unlike other methods that gave the same weight if samples have at least one similar label between them during training, it can be seen that our method shows the best results by a large margin which shows that our method addressed the ranking problem well by exploiting the label information fully.

Comparisons with Current Deep Cross-Modal Hashing Methods: We also compared our method with current deep cross-modal hashing methods as shown in Table 3.[10] It can be seen, that our model gave best results, using the shared binary code as gallery for the two benchmark datasets. This may be due to several reasons; First, the CAH method still used handcrafted image features as input for their deep networks while our method performed a complete network learning from raw images. Second, the DCMH method performed end-to-end learning but exploited the label information directly to the neural code output of the hash networks, and not the binary code which may have lead to some approximation loss. Finally, DVSH and DNH-C both performed end-to-end supervised metric-based network training in the form of cosine hinge loss and triplet ranking loss, respectively, which may not fully obtain discriminative binary codes compared to our classification-based hinge loss learning.

Empirical Analysis: We also investigated variants of our CMDVH method to see the importance of each aspect of our architecture and learning method. CMDVH1 ignores the latent network in the cross-modal fusion network which assumes that simply combining the outputs of the image and text network would be representative enough for binary code inference. CMDVH2 ignores the probabilistic interpretation of the modality-specific network and simply learn the binary codes from a negative log likelihood loss. Table 4 shows the performance of these variants on the NUS-WIDE database. We see that a fusion network is still important to perform the nonlinear transformation to make the learned codes more representative.

Table 4. mAP performance of different variants of our CMDVH method on the NUS-WIDE dataset.

	Method	16 bits	32 bits	64 bits	128 bits
$I \to T$	CMDVH1	0.7864	0.8615	0.8631	0.8666
	CMDVH2	0.8234	0.8576	0.8762	0.8821
	CMDVH	**0.8503**	**0.8755**	**0.8801**	**0.8910**
$T \to I$	CMDVH1	0.7390	0.8206	0.8375	0.8504
	CMDVH2	0.7992	0.8280	0.8282	0.8583
	CMDVH	**0.8270**	**0.8328**	**0.8403**	**0.8782**

5. Conclusion

In this paper, we have proposed a cross-modal deep variational hashing (CMDVH) for scalable multimedia retrieval. Our method learns a fusion network to learn binary codes from cross-modal training pairs which exploits class label information, which learn a generative modality-specific hash network for the out-of-sample extension. Experimental results on three multimedia retrieval datasets have shown the effectiveness of the proposed approach.

Acknowledgements

This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1001001 and the National Natural Science Foun-

[10] Results are obtained from the respective author's papers. We used the same experimental setup as mentioned in their papers.

dation of China under Grant 61672306. This research was also carried out at the Rapid-Rich Object Search (ROSE) Lab at the Nanyang Technological University, Singapore. The ROSE Lab is supported by the Infocomm Media Development Authority, Singapore

References

[1] M. M. Bronstein, A. M. Bronstein, F. Michel, and N. Paragios. Data fusion through cross-modality metric learning using similarity-sensitive hashing. In *CVPR*, pages 3594–3601, 2010. 2, 3, 6

[2] Y. Cao, M. Long, J. Wang, Q. Yang, and P. S. Yu. Deep visual-semantic hashing for cross-modal retrieval. In *KDD*, pages 1–10, 2016. 3, 6, 7

[3] Y. Cao, M. Long, J. Wang, and H. Zhu. Correlation autoencoder hashing for supervised cross-modal search. In *ICMR*, pages 197–204, 2016. 3, 7

[4] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014. 6

[5] G. Ding, Y. Guo, and J. Zhou. Collective matrix factorization hashing for multimodal data. In *CVPR*, pages 2083–2090, 2014. 2, 3, 7

[6] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *ICAIS*, pages 249–256, 2010. 7

[7] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik. A multi-view embedding space for modeling internet images, tags, and their semantics. *IJCV*, 106(2):210–233, 2014. 2

[8] Y. Gong and S. Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *CVPR*, pages 817–824, 2011. 1, 7

[9] K. Jiang, Q. Que, and B. Kulis. Revisiting kernelized locality-sensitive hashing for improved large-scale image retrieval. In *CVPR*, pages 1–10, 2015. 1

[10] Q.-Y. Jiang and W.-J. Li. Deep cross-modal hashing. *arXiv preprint arXiv:1602.02255*, pages 1–12, 2016. 3, 7

[11] C. Kang, S. Xiang, S. Liao, C. Xu, and C. Pan. Learning consistent feature representation for cross-modal multimedia retrieval. *TMM*, 17(3):370–381, 2015. 2

[12] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *ICLR*, pages 1–14, 2014. 4, 5

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 3

[14] S. Kumar and R. Udupa. Learning hash functions for cross-view similarity search. In *IJCAI*, volume 22, pages 1360 – 1365, 2011. 2, 6, 7

[15] H. Lai, Y. Pan, Y. Liu, and S. Yan. Simultaneous feature learning and hash coding with deep neural networks. In *CVPR*, pages 3270–3278, 2015. 7

[16] C. Leng, J. Wu, J. Cheng, X. Bai, and H. Lu. Online sketching hashing. In *CVPR*, pages 2503–2511, 2015. 1

[17] Z. Lin, G. Ding, M. Hu, and J. Wang. Semantics-preserving hashing for cross-view retrieval. In *CVPR*, pages 3864–3872, 2015. 2, 3, 7

[18] W. Liu, C. Mu, S. Kumar, and S.-F. Chang. Discrete graph hashing. In *NIPS*, pages 3419–3427, 2014. 4

[19] J. Masci, M. M. Bronstein, A. M. Bronstein, and J. Schmidhuber. Multimodal similarity-preserving hashing. *TPAMI*, 36(4):824–830, 2014. 3

[20] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814, 2010. 6

[21] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *ICML*, pages 689–696, 2011. 3

[22] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos. A new approach to cross-modal multimedia retrieval. In *ACM MM*, pages 251–260, 2010. 2

[23] M. Rastegari, J. Choi, S. Fakhraei, D. Hal, and L. Davis. Predictable dual-view hashing. In *ICML*, pages 1328–1336, 2013. 6, 7

[24] F. Shen, C. Shen, W. Liu, and H. Shen. Supervised discrete hashing. In *CVPR*, pages 37–45, 2015. 1

[25] C. Szegedy, A. Toshev, and D. Erhan. Deep neural networks for object detection. In *NIPS*, pages 2553–2561, 2013. 3

[26] A. Vedaldi and K. Lenc. Matconvnet: Convolutional neural networks for matlab. In *ACM MM*, pages 689–692, 2015. 6

[27] C. Wang, H. Yang, and C. Meinel. A deep semantic framework for multimodal representation learning. *Multimedia Tools and Applications*, pages 1–22, 2016. 3

[28] D. Wang, X. Gao, X. Wang, and L. He. Semantic topic multimodal hashing for cross-media retrieval. In *IJCAI*, pages 3890–3896, 2015. 2

[29] J. Wang, S. Kumar, and S.-F. Chang. Semi-supervised hashing for large-scale search. *TPAMI*, 34(12):2393–2406, 2012. 1

[30] K. Wang, R. He, L. Wang, W. Wang, and T. Tan. Joint feature selection and subspace learning for cross-modal retrieval. *T-PAMI*, 38(10):2010–2023, 2016. 2

[31] K. Wang, R. He, W. Wang, L. Wang, and T. Tan. Learning coupled feature spaces for cross-modal matching. In *ICCV*, pages 2088–2095, 2013. 2

[32] W. Wang, X. Yang, B. C. Ooi, D. Zhang, and Y. Zhuang. Effective deep learning-based multi-modal retrieval. *VLDB*, 25(1):79–101, 2016. 3

[33] Y. Wei, Y. Zhao, C. Lu, S. Wei, L. Liu, Z. Zhu, and S. Yan. Cross-modal retrieval with cnn visual features: A new baseline. *TSCVT*, 47(2):449–460, 2017. 3

[34] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *NIPS*, pages 1753–1760, 2008. 1

[35] X. Xu, F. Shen, Y. Yang, and H. T. Shen. Discriminant cross-modal hashing. In *ICMR*, pages 305–308, 2016. 2, 3, 7

[36] D. Zhang and W.-J. Li. Large-scale supervised multimodal hashing with semantic correlation maximization. In *AAAI*, pages 2177–2183, 2014. 2, 3, 7

[37] J. Zhou, G. Ding, and Y. Guo. Latent semantic sparse hashing for cross-modal similarity search. In *ACM SIGIR*, pages 415–424, 2014. 2, 3, 7

[38] X. Zhu, Z. Huang, H. T. Shen, and X. Zhao. Linear cross-modal hashing for efficient multimedia search. In *ACM MM*, pages 143–152, 2013. 2