Learning Compact Hash Codes for Multimodal Representations using Orthogonal Deep Structure

Daixin Wang, Peng Cui, Mingdong Ou, and Wenwu Zhu, *Fellow, IEEE*

Abstract—As large-scale multimodal data are ubiquitous in many real-world applications, learning multimodal representations for efficient retrieval is a fundamental problem. Most existing methods adopt shallow structures to perform multimodal representation learning. Due to a limitation of learning ability of shallow structures, they fail to capture the correlation of multiple modalities. Recently, multimodal deep learning was proposed and had proven its superiority in representing multimodal data due to its high nonlinearity. However, in order to learn compact and accurate representations, how to reduce the redundant information lying in the multimodal representations and incorporate different complexities of different modalities in the deep models is still an open problem. In order to address the aforementioned problem, we propose a hashing-based orthogonal deep model to learn accurate and compact multimodal representations in this paper. The method can better capture the intra-modality and inter-modality correlations to learn accurate representations. Meanwhile, in order to make the representations compact, the hashing-based model can generate compact hash codes and the proposed orthogonal structure can reduce the redundant information lying in the codes by imposing orthogonal regularizer on the weighting matrices. We also theoretically prove that in this case the learned codes are guaranteed to be approximately orthogonal. Moreover, considering the different characteristics of different modalities, effective representations can be attained with different number of layers for different modalities. Comprehensive experiments on three real-world datasets demonstrate a substantial gain of our method on retrieval tasks compared with existing algorithms.

Index Terms—Multimodal hashing, Deep learning, Similarity Search

1 Introduction

Nowadays, huge volumes of multimodal contents have been generated on the Internet, such as texts, videos and images. These multimodal data carry different kinds of information, and these multimodal information is often needed to integrate to get comprehensive results in many real-world applications. For example, image tagging aims to find relevant tags for images, recommendation systems aim to find preferred multimodal items (e.g. web posts with texts and images) for users, and image search aims to retrieve images for text queries. Among these important applications, how to learn multimodal representations for efficient similarity search is a fundamental problem.

As [1] claims, compactness and accuracy are two important factors for good representations. The key problem for an accurate representation for multimodal data is to capture the correlation of multiple modalities, because they can well model the posterior distribution of the observed multimodal input data. Meanwhile, the key problem for a compact representation is dependent on the feature dimension and discreteness [2]. Compact representation can be processed efficiently. To date, many machine learning methods have been proposed to learn good representations for multimodal data, such as Support Vector Machine (SVM) [3], Independent Component Analysis (ICA) [4], Principle Component Analysis (PCA) [5]. However, most of existing methods are shallow structures. As [6] proved, correlation between different modalities exists in the high-level space and the mapping functions from raw feature space to high level space are highly nonlinear. Thus, it is difficult for shallow models to learn such high-level correlation [7] to get an accurate multimodal representation. Nowadays, deep learning [8] has been proved to be able to discover high-level representation for visual [9], textual [10] modality. It has multiple layers and each layer is always a non-linear transformation to map low-level features to high level space [11]. Motivated by it, multimodal deep learning [6] [12] [13] has been proposed to capture the high-level correlation in multimodal information. Despite their success in modeling the posterior distributions of multimodal input data, these methods all learn high-dimensional real-valued latent features to represent multimodal data. However, faced with such a large amount of multimodal data, such representations are not compact to perform retrieval task.

To learn compact representations, hashing is an effective way to reduce the dimension of the representations and discretize the representations [14]. It maps objects represented by high-dimensional raw features into Hamming space, where the objects are represented by short binary hash codes. When retrieving samples, hashing can implement efficient search by searching for the samples whose hash codes are within a small Hamming distance of the hash

- *Daixin Wang, Peng Cui, Mingdong Ou and Wenwu Zhu are with the Department of Computer Science and Technology, Tsinghua University, China.*

codes of the query. In addition, the compact hash codes can dramatically reduce the storage. Therefore, it is desired to incorporate deep learning with hashing method to learn compact and accurate representations for multimodal data. [15] has proposed a multimodal deep learning scheme to perform hashing. However, existing works still face the following challenges. Firstly, there exists redundancy in the learned representations, which makes the final representation incompact and imprecise and seriously affects the performance. In shallow structures, it is common to impose orthogonality constraint on the global dataset to decorrelate different bits. For deep model, since it is highly non-linear, its objective function is non-convex, which makes many local minima in the model parameter space [11]. In order to alleviate this problem, deep learning often adopts mini-batch learning rather than the global dataset to do gradient descent [16]. But this intrinsically prohibits the possibility to directly impose orthogonality constraint on the global dataset. Thus, how to solve the redundancy problem is a challenging task. Secondly, the performance for retrieval usually decreases when the representation becomes compact. Therefore, how to make tradeoffs between compactness and accuracy is also challenging. Finally, most of the previously proposed deep models adopt symmetric structures, with the assumption that different modalities possess the same complexity. However, it is intuitive visual data has much larger semantic gap than that of textual data, which results in different complexities in different modalities. How to address the imbalanced complexity problem in deep learning models is also critical for multimodal hashing.

To address the above challenges, we propose a hashing-based orthogonal deep model to learn compact binary codes for multimodal representations. The model can fully exploit intra-modality and inter-modality correlations to learn accurate representations. To address the redundancy challenge lying in the binary codes, we impose orthogonal regularizer on the weighting matrices of the deep structure, and theoretically prove that in this case the learned representation is approximately guaranteed to be orthogonal. Then we optimize the compactness and accuracy together in a unified model to make a good trade-off between them. Furthermore, considering different characteristics of different modalities, the proposed modality-specific structures can learn more effective representations by assigning different number of layers for different modalities.

The overall contributions of our paper are listed as follows:

- We propose a novel deep learning framework to generate compact and accurate hash codes for multimodal data by exploiting both the intra-modality and inter-modality correlation and incorporating different complexities of different modalities.
- We propose a novel method with theoretical basis to reduce the redundancy in the learned hashing representation by imposing orthogonal regularization on the weighting parameters.
- Experiments on three real-world datasets demonstrate a substantial gain of our model compared to other state-of-the-art hashing methods.

2 Related work

There have been many methods focusing on learning representations for multimodal data. [17] demonstrates that incorporating tags or captions with images' low-level features can help multiple kernel learning improve the performance of image classification. Similarly, [18] uses image features and corresponding textual description to help improve the performance of SVM. [19] integrates visual features into LDA and proposes a multimodal LDA to learn representations for textual and visual data. However, these work do not focus on dealing with the compactness aspect of representations. They mainly generate high-dimensional real-valued features to do classification. To perform retrieval task, compact representations are very critical. To learn compact representations with little sacrifice on accuracy, hashing is a widely used method. In general, there are mainly two different ways for hashing-based similarity search to generate hash codes, i.e. data-independent and data-dependent ways. The difference between these two methods lies in the method of generating the hash function.

Data-independent hashing methods often generate random projections as hash functions. Locality Sensitive Hashing (LSH) [20] is one of the most well-known representative. It uses a set of random locality sensitive hash functions to map examples to hash codes. Along this direction further improvements like multi-probe LSH [21] [22] are proposed but the performance is still limited by the random technic. Unlike these approaches which randomly project the data into Hamming space, data-dependent hashing methods adopt machine learning method to perform the projection. Such methods utilize the distribution of data or some supervised information to help improve the retrieval quality. Spectral Hashing [23] is a representative of machine learning based hashing. The method assumes data are distributed uniformly in a hyper-rectangle and uses graph laplacian for finding optimal hash codes. However, such assumption is not reasonable in most cases. In [24], the authors use a deep learning based model to do hashing, called semantic hashing. [25], [26] follow this idea and extend the multi-layer structure to do image search on a large image database. These two approaches are both unsupervised methods and make use of the powerful representation ability of deep learning method. Besides unsupervised hashing, supervised learning is also adopted in hashing-based methods. For example, [14] uses latent SVM, [27] adopts kernel method and [28] implements metric learning based hashing. To utilize both the labeled information and data's own distribution, [29] proposes a semi-supervised scheme which can both minimize the empirical error on the labeled data and provide effective regularization.

However, most of the above methods are designed for single modality features. But in the real-world we always need to process information with multiple modalities. The problem of multimodal hashing is first proposed by CMSSH [30]. The work treats the mapping procedure as a binary classification and the function is learnt by using boosting algorithm. CMSSH shows its superiority compared with single modality hash method. But it only considers the inter-modality correlation and ignores intra-modality correlation. Some further works focusing on encoding ex-

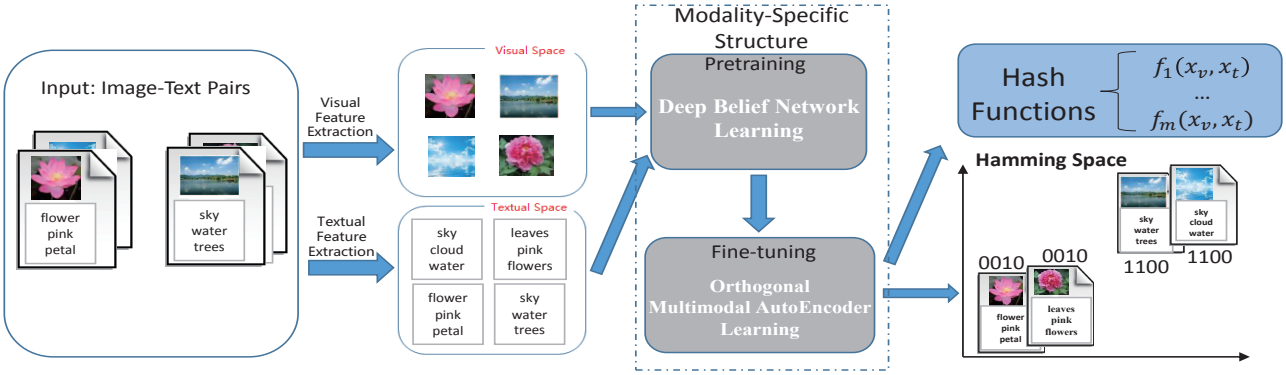


Fig. 1: Concept diagram of the our proposed multimodal hashing model.

amples represented by multimodal features are proposed [31] [32] [33] [34] [35] [36]. For example, CVH [37] extends spectral hashing to multiview and leverages intra-modality correlation and inter-modality correlation together. PDH [38] applies max-margin theory to do multimodal hashing. RAHH [39] incorporates heterogeneous relationship between different modalities to help learn multimodal hash function. However, these multimodal hash models adopt shallow-layer structures. It is hard for them to map low-level features to a high-level space to capture the correlation between different modalities.

To capture the correlation between multiple modalities, a few recent works focusing on multimodal deep learning are proposed. [6] [12] [13] target at learning high-dimensional latent features to perform discriminative classification task. [40] [41] apply autoencoder to perform cross-modality retrieval. However, these methods deal with the different task from our task of learning compact binary codes for multimodal data. From the angle of targeted problem, the most related work with ours is [15], which proposed a multimodal deep learning scheme to perform hashing. However, in their scheme, it does not consider the redundant information between different bits of hash codes. The redundancy in hash codes can badly influence the performance of similarity search due to the compact characteristic of hash representations. In addition, they fail to consider the different complexity of different modalities.

3 Modality-specific Deep Multimodal Hashing

In this section, we first give an introduction of the multimodal hashing task. Then we define the terms and notation for our framework and further formulate the problem of multimodal hashing. Finally, we introduce the corresponding framework of Modality-specific Deep Multimodal Hashing.

3.1 The Introduction of the Multimodal Hashing Task

Firstly, we give a brief introduction of the concept diagram of our multimodal hashing task as shown in Figure 1. Without loss of generality, to describe our task we assume image and text are two modalities. Then in the section of model formulation and problem soluation, we will extend our model into multiple modalities.

The input are images with their textual descriptions. The textual descriptions can be words, sentences or articles. We first extract the image features and text features for the image-text pairs. Then a Deep Belief Network is adopted to do pretraining. After pretraining, an Orthogonal Multimodal Autoencoder is proposed to fine-tune the parameters of the pretraining phase. After the above two phases, we get the hash function and we can use it to map examples to the Hamming space to get corresponded binary hash codes. Next, we will introduce the model in detail.

3.2 Notation and Problem Statement

The terms and notation of our model are listed in Table 1. Note that the subscript p represents the p-th modality. The superscript l denotes the l-th layer.

TABLE 1: Terms and Notation

Symbol	Definition
m_p	number of hidden layers in the p-th pathway
n	the number of samples
m	the length of the hash codes
P	the number of modalities
$\mathbf{x}_p$	input of the p-th modality
$\mathbf{z}_p^{(l)}$	representations of the l-th hidden layer for the p-th modality
$\mathbf{z}$	representations of the top joint layer
$\mathbf{h}$	hash codes
$W_p^{(l)}$	the l-th layer's weight matrix for the p-th modality
$\mathbf{b}_p^{(l)}$	the l-th up-biases for the p-th modality
$\mathbf{c}_p^{(l)}$	the l-th down-biases for the p-th modality
θ	$\{W_p^{(l)}, \mathbf{b}_p^{(l)}, \mathbf{c}_p^{(l)}\}_{l,p}$
$s_p^{(l)}$	the number of l-th layer's units for the p-th modality

Suppose that we have n training examples X from P modalities. The data of each modalities are represented by X_p. The main objective is to find a hash function $\mathbf{f} : (\mathbb{R}^{d_1}, ..., \mathbb{R}^{d_P}) \mapsto \mathbb{H}^m$, where d_p is the dimensionality of X_p, $\mathbb{H}^m$ represents the Hamming space of dimension m. If two objects o_1 and o_2 are semantic similar, the hash functions should satisfy that the distance of hash codes $f(o_1)$ and $f(o_2)$ is small. After learning the hash function, given any sample denoted as $\mathbf{x} = \{\mathbf{x}_p\}_{p=1}^P$, its corresponding hash codes $\mathbf{h}$ are calculated as $\mathbf{h} = f(\mathbf{x}_1, ..., \mathbf{x}_P)$.

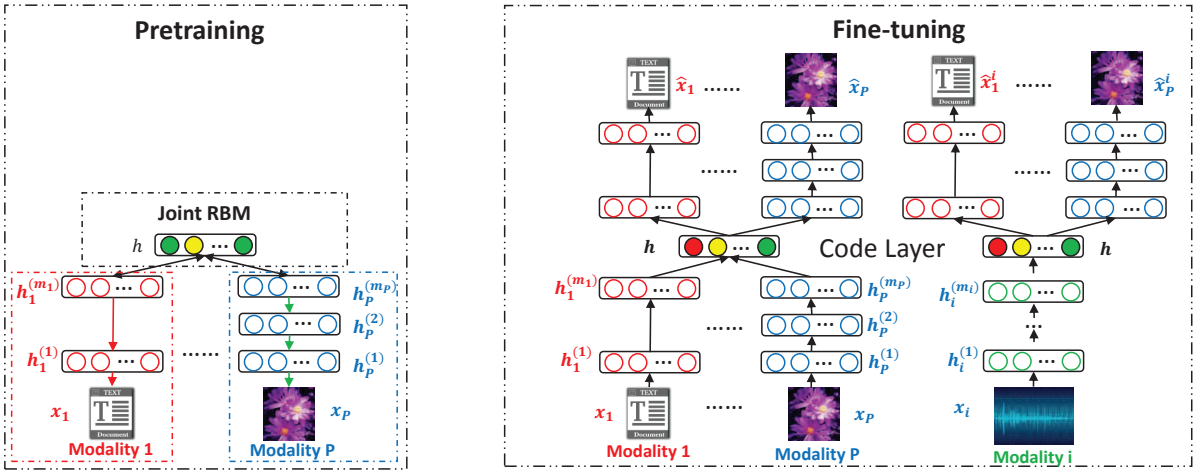


Fig. 2: Our framework (a) The multimodal DBN in pretraining (b) The multimodal AutoEncoder and the cross-modality Autoencoder in fine-tuning. They are optimized together. The same color in different bits means they may contain redundant information.

3.3 The Framework

3.3.1 Pretraining

Due to the high nonlinearity, deep-structured models often suffer from many local minima in the parameter space. In order to find a good region of parameter space, we first adopt pretraining in our multimodal hashing.

As the correlation of multiple modalities exists in the high-level space, we employ a multimodal Deep Belief Network (mDBN) as shown in Figure 2(a). The mDBN is composed of multiple seperate DBNs and a joint Restricted Boltzmann Machine (RBM). The multiple DBNs, each of which represents one modality, map individual low-level features to high-level space and the joint RBM captures the correlation of multiple modalities.

In detail, every two adjacent layers in a DBN form a Restricted Boltzmann Machine (RBM) [7] [8]. The RBM is a bipartite graph between the hidden layer and visible layer. The visible layer denotes the observed data and the hidden layer denotes the latent variables that generate the data. The objective function of the RBM is to maximize the generative probability of the observed data. For different input types, we choose different types of RBM, such as basic RBM [42] for 0,1 vector, Gaussian RBM [43] for real-valued data, Replicated Softmax RBM [44] for word-count data. For basic RBM, let $\mathbf{v} = \{v_i\}_{i=1}^{M} \in \{0,1\}^M$ denote the visible units and $\mathbf{h} = \{h_j\}_{j=1}^{N} \in \{0,1\}^N$ denote the hidden units, where M denotes the number of visible units, N denotes the number of hidden units. Its conditional probabilities are as follows:

$$\begin{aligned} p(h_j = 1|\mathbf{v}) &= \sigma(c_j + \sum\nolimits_i w_{ij} v_i) \\ p(v_i = 1|\mathbf{h}) &= \sigma(b_i + \sum\nolimits_j w_{ij} h_j), \end{aligned} \tag{1}$$

where σ denotes the logistic sigmoid function, w_{ij} denotes the weights between v_i and h_j, b_i and c_j denote the bias terms.

For other types RBM, we can also get similar conditional probabilities. To learn RBM, exact learning is intractable [42]. Thus, we adopt 1-step Contrastive Divergence [45] to do approximate learning of RBM by using the above conditional probabilities. Then we can go even further to learn higher level representation by adding more layers on the RBM to form a DBN. To train the DBN, we adopt the method of greedy layer-wise training [8].

After using modality-specific DBN to get the high-level representation for each modality, we cannot use it to get hash codes directly because different modalities have different bias, making them incomparable across different modalities. Therefore, to find the common representation, we add an additional layer on top of the two DBNs, forming a joint RBM. The conditional probabilities of joint RBM are defined as:

$$P(\mathbf{z}|\mathbf{z}_1^{(m_1)}, ..., \mathbf{z}_P^{(m_P)}) = \sigma\Big(\sum_{1 \le p \le P} (W_p^{(m_p+1)T} \mathbf{z}_p^{(m_p)} + \mathbf{b}_p^{(m_p+1)})\Big)$$
$$P(\mathbf{z}_p^{(m_p)}|\mathbf{z}) = \sigma(W_p^{(m_p+1)T} \mathbf{z} + \mathbf{c}_p^{(m_p+1)}).$$

Similarly, we apply Contrastive Divergence to train the joint RBM. Since deep model has many parameters, we adopt dropout training [46] over the entire network in the overall pretraining phase to prevent overfitting.

3.3.2 Fine-Tuning

After pretraining, the parameters lie in a good region of the parameter space which is near the right neighborhood [25]. But it is not optimal. Thus, we do fine-tuning to refine the parameters by performing local gradient search to move the parameters to the local optimum.

To learn accurate representations, we need to incorporate both intra-modality and inter-modality correlation to extract more discriminative information. To preserve the intra-modality correlation, we unroll the mDBN to form the multimodal Autoencoder (MAE) as shown in the left part of Figure 2(b). Given input of multiple modalities, the joint representation is demanded to reconstruct all modalities.

The loss function of any pair of input $\mathbf{x} = \{\mathbf{x}_p\}_{p=1}^P$ is defined as follows:

$$L_{MAE}(\mathbf{x};\theta) = \frac{1}{2}(\sum_{p=1}^{P} \|\hat{\mathbf{x}}_p - \mathbf{x}_p\|_2^2), \tag{2}$$

where $\hat{\mathbf{x}}_p$ is the reconstruction of $\mathbf{x}_p$.

By minimizing the above reconstruction loss function, the hidden representations can well reconstruct the input data, which ensures that they maintain the intra-modality correlation. However if we only maintain the intra-modality correlation, some units of the joint representation may contain only modality-specific information which is far less discriminative than the common information. Therefore, we also need to preserve the inter-modality correlation. Inspired by [12], we further propose a cross-modality Autoencoder (CAE) as shown in right part of Figure 2(b). In this model, when only one modality is present and the rest are absent, the learned representation is still required to be able to reconstruct all the modalities. In this way, the common information for all modalities is strengthened and the modality-specific information is weakened, which results in the effect of capturing the inter-modality correlation.

For our model, since we have P modalities, we can use any one of the modalities as input to form a CAE. Thus, there overall exists P CAE. We use p-modality-only CAE to denote the CAE which only uses the data of the p-th modality X_p as input. The loss function for p-modality-only CAE of input $\mathbf{x} = \{\mathbf{x}_p\}_{p=1}^P$ is defined as follows:

$$L_{CAE}^p(\mathbf{x};\theta) = \frac{1}{2}(\sum_{q=1}^{P} \|\hat{\mathbf{x}}_q^p - \mathbf{x}_q\|_2^2), \tag{3}$$

where the superscript p of L_{CAE}^p denotes the p-modality-only CAE. $\hat{\mathbf{x}}_q^p$ is the reconstruction of input of q-th modality $\mathbf{x}_q$ in p-modality-only CAE. To calculate $\hat{\mathbf{x}}_q^p$, we set the missing modality to zero in the joint layer. The specific mathematical expressions of above notations are defined in Section 3.3.4.

3.3.3 Modality-specific Structure

Even if different modalities describe the same object, they still have different statistical properties in the low-level raw feature space. The high correlation exists in the high-level semantic space. Thus, deep-structured models adopt multiple layers to map low-level raw feature space to high-level space to find such correlation. However, the gap between low-level feature space to high-level semantic space varies for different modalities. For example, the gap between visual pixels and object categories is much larger than the gap between textual words to topics, which means that the visual modality possess higher complexity than text modality. Therefore, we propose a modality-specific structure on all of the above models to incorporate different complexities of different modalities. In particular, we endow the model with the flexibility of varying the number of layers for different modalities independently. The experimental results clearly demonstrate that the optimal solution is achieved with different number of layers for different modalities.

3.3.4 Hash Function

To define the hash function, we first define the representation of the l-th hidden layer $\mathbf{z}_p^{(l)}$, $1 \le p \le P$ for each modality, and the joint representation $\mathbf{z}$ as follows:

$$\begin{aligned}
\mathbf{z}_p^{(1)} &= \sigma(W_p^{(1)T}\mathbf{x}_p + \mathbf{b}_p^{(1)}) \\
\mathbf{z}_p^{(l)} &= \sigma(W_p^{(l)T}\mathbf{z}_p^{(l-1)} + \mathbf{b}_p^{(l)}), l = 2, ..., m_p \\
\mathbf{z} \quad &= \sigma\Big[\sum_{1\le p\le P}(W_p^{(m_p+1)T}\mathbf{z}_p^{(m_p)} + \mathbf{b}_p^{(m_p+1)})\Big],
\end{aligned} \tag{4}$$

where $\sigma(\cdot)$ is the sigmoid function. Specifically for CAE, we set the missing modality to zero when calculating Eq. 4.

For the decoder part of CAE or MAE, the representation is calculated reversely in the following ways.

$$\begin{aligned}
\hat{\mathbf{z}}_p^{(m_p)} &= \sigma(W_p^{(m_p+1)T}\mathbf{z} + \mathbf{c}_p^{(m_p+1)}) \\
\hat{\mathbf{z}}_p^{(l)} &= \sigma(W_p^{(l+1)T}\hat{\mathbf{z}}_i^{(l+1)} + \mathbf{c}_p^{(l+1)}), l = 2, ..., m_p - 1 \\
\hat{\mathbf{x}}_p \quad &= \sigma(W_p^{(2)T}\hat{\mathbf{z}}_p^{(2)} + \mathbf{c}_p^{(2)}).
\end{aligned}$$

Then the hash function $f(\cdot)$ and hash codes $\mathbf{h}$ for input $\mathbf{x}_i$ are defined as follows:

$$\mathbf{h} = f(\mathbf{x}_1, ..., \mathbf{x}_P; \theta) = \mathbf{I}\big[\mathbf{z} \ge \delta\big] \in \{0,1\}^m, \tag{5}$$

where $\mathbf{I}$ is the indicator function and δ is the threshold.

To obtain a hash function, we need to learn optimal parameters to both preserve intra-modality and inter-modality correlation. Thus, a straightforward solution to the hash function is to jointly optimize the MAE and all the p-modality-only CAE together. The formulation to the optimization problem is then proposed as follows:

$$\min_{\theta} \quad L_1(X;\theta) = \frac{1}{n}\sum_{i=1}^{n}(L_{MAE} + \sum_{p=1}^{P}\lambda_p L_{CAE}^p) + \mu L_{reg},$$

where L_{reg} is an $\mathcal{L}2$-norm regularizer term of the weight matrix which is defined as follows:

$$L_{reg} = \frac{1}{2}(\sum_{p=1}^{P}\sum_{l=1}^{m_p+1}\|W_p^{(l)}\|_F^2). \tag{6}$$

The equation can decrease the magnitude of the weights and is commonly used to prevent overfitting [47].

4 Orthogonal Regularized Deep Structure for Learning Compact Representations

Above section mainly maintains the discriminability aspect of the representations. But it is not enough for good representations. We should also take compactness into account. Especially for hash representation learning, compactness is a critical criterion to guarantee its performance in efficient similarity search. Given a certain small length of binary codes, the redundancy existing between different bits would badly affect the performance of compact representations. By removing the redundancy, we can either incorporate more information in the same length of binary codes, or shorten the binary codes while maintaining the same amount of information. Therefore, we propose the following Orthogonal Regularized Deep Structure to alleviate the redundancy problem to learn compact representations.

4.1 Formulation

To alleviate the redundancy problem lying in different bits, we impose the orthogonality constraints to decorrelate different bits. Since H is non-negative as is shown in Eq. 5, we first transform H to $\tilde{H} = 2H - \mathbf{1}$. In this way, any element in $\tilde{H}$ is either -1 or 1. Then we formulate the problem as the following objective function:

$$\begin{aligned} \min_{\theta} \quad & L_1(X;\theta) = \frac{1}{n}\sum_{i=1}^{n}(L_{MAE} + \sum_{p=1}^{P}\lambda_p L_{CAE}^p) + \mu L_{reg} \\ s.t. \quad & \frac{1}{n}\tilde{H}^T \cdot \tilde{H} = I. \end{aligned} \tag{7}$$

The above objective function is a hard problem in two aspects. Firstly, the value of $\tilde{H}$ is discrete, which makes $\tilde{H}$ non-differentiable. To solve it, we follow [23] to remove the discrete constraint. Secondly, the orthogonality constraint constrains the hash codes of the global dataset. However, mini-batch training is commonly adopted for deep learning. Thus, it prevents us directly solving the optimization problem which constrains on the global dataset. To solve this, we provide the following two lemmas to help transform the objective function.

Lemma 4.1. *Suppose $Z = \sigma(WX^T)$, if X,W are orthogonal matrix. Then the matrix $\tilde{Z} = 2Z - 1$ retains orthogonality.*

Proof. The one-order Taylor Series of Z at $X = 0$ is:

$$\begin{aligned} Z &\approx Z(0) + \frac{\partial Z(0)}{\partial X}X^T \\ &= \frac{1}{2} + \frac{1}{4}WX^T. \end{aligned}$$

Therefore, $\tilde{Z} = 2Z - 1 = \frac{1}{2}WX^T$.

$$\tilde{Z}^T \cdot \tilde{Z} \propto XW^TWX^T \propto I.$$

□

Lemma 4.2. *Suppose $H = \sigma(\sum_p W_p X_p^T)$. If X_p,W_p are orthogonal matrices, $W_p^T W_q = 0$ $(p \neq q)$, then the matrix $\tilde{H} = 2H - 1$ satisfies $\tilde{H}^T \cdot \tilde{H} \propto I$.*

Proof. The one-order Taylor Series of H at $X_p = 0$ is:

$$\begin{aligned} H &\approx H(0,0) + (\sum_p \frac{\partial}{\partial X_p}X_p^T)H(0,0) \\ &= \frac{1}{2} + \frac{1}{4}(\sum_p W_p X_p^T). \end{aligned}$$

Therefore, $\tilde{H} = 2H - 1 = \frac{1}{2}(\sum_p W_p X_p^T)$.

$$\begin{aligned} \tilde{H}^T \cdot \tilde{H} &\propto \sum_p X_p W_p^T W_p X_p^T + \sum_{p,q,p\neq q} X_p W_p^T W_q X_q^T \\ &\propto I. \end{aligned}$$

□

Under the assumption that input X_p are orthogonal [29], if we impose the orthogonality constraints on each layer's weight matrix as shown in Eq.8, the Lemma 4.1 guarantees that the representations $\mathbf{z}_p^{(m_p)}$ are approximately orthogonal. If we further impose the constraint of Eq. 9, the Lemma 4.2 guarantees the orthogonality of the hash codes. Thus we can impose the orthogonal regularization on the weighting matrices instead of the hash bits, which significantly facilitate the optimization process. Therefore, we propose the following new objective function:

$$\begin{aligned} \min_{\theta} \quad & L_1 \\ s.t. \quad & {W_p^{(l)}}^T \cdot W_p^{(l)} = I, \quad l = 1, ..., m_p + 1 \quad , 1 \leq p \leq P \quad (8) \\ & W_p^{(m_p+1)} \cdot {W_q^{(m_q+1)}}^T = 0, \quad 1 \leq p, q \leq P, \quad p \neq q. \quad (9) \end{aligned}$$

Inspired by [29], the hard orthogonality constraints may reduce the quality. Thus, instead of imposing hard orthogonality constraints, we add penalty terms on the objective function and propose the following final overall objective function:

$$\begin{aligned} \min_{\theta} \quad L(X;\theta) = L_1 &+ \sum_{p=1}^{P}\sum_{l=1}^{m_p+1}\alpha_l^p \|{W_p^{(l)}}^T W_p^{(l)} - I\|_F^2 \\ &+ \sum_{p,q}\beta_{p,q}\|W_p^{(m_p+1)} \cdot {W_q^{(m_q+1)}}^T\|_F^2. \end{aligned} \tag{10}$$

By minimizing the above objective function, the joint representations can not only preserve the intra-modality and inter-modality correlation by minimizing the reconstruction error of MAE and CAE, but we also alleviate the redundancy problem lying in different bits. In this case, the learned representations are both compact and accurate.

4.2 Solution

We adopt back-propagation on Eq. 10 to fine-tune the parameters. We calculate the derivative of $W_p^{(m_p+1)}$ as an example [1]:

$$\frac{\partial L}{\partial W_p} = \frac{\partial L_1}{\partial W_p} + \alpha_{m_p+1}^p \frac{\partial \|W_p^T W_p - I\|_F^2}{\partial W_p} + \sum_q \beta_{p,q}\frac{\partial \|W_p W_q^T\|_F^2}{\partial W_p}. \tag{11}$$

The calculation of the first term is the same as most basic autoencoders. The second and third terms are calculated as follows:

$$\begin{aligned} \frac{\partial \|W_p^T W_p - I\|_F^2}{\partial W_p} &= \frac{\partial \mathbf{tr}[(W_p^T W_p - I)^T (W_p^T W_p - I)]}{\partial W_p} \\ &= 4 \times (W_p W_p^T - I)W_p. \\ \frac{\partial \|W_p W_q^T\|_F^2}{\partial W_p} &= \frac{\partial \mathbf{tr}[(W_p W_q^T)^T (W_p W_q^T)]}{\partial W_p} = 2W_p W_q^T W_q \end{aligned} \tag{12}$$

The update of other parameters follows a similar way. The fine-tuning algorithm is presented in Algorithm 1.

After finishing training all of the parameters, we can use Eq.5 to derive representations for any samples. We use the median value of all the training samples as the threshold δ to perform binarization and generate hash codes.

4.3 Complexity Analysis

The overall complexity is composed of training complexity and online testing complexity. For testing complexity, the calculation of the hash codes can be performed using a few matrix multiplications, which is fast and is linear with the

1. Here, for simplicity, $W_p^{(m_p+1)}$ and $W_q^{(m_q+1)}$ is simply denoted as W_p and W_q, $p \neq q$

Algorithm 1 Fine-tuning

Input: $X = \{X_p\}_{p=1}^{P}$, θ;
Output: New Parameters: θ

repeat
  for batch $B = \{B_p\}_{p=1}^{P}$ in $\{X_p\}_{p=1}^{P}$ **do**
    Apply Eq.3 to get $L_{CAE}^{p}(B;\theta)$.
    Apply Eq.2 to get $L_{MAE}(B;\theta)$.
    Apply Eq.10 to get $L(B;\theta)$
    Use $\partial L(B;\theta)/\partial\theta$ to back-propagate through the entire network to get new θ
  end for
until converge

number of query data and irrelevant with the size of training data [24].

For pretraining, we suppose that each RBM is pretrained for k_1 epochs. Thus, the computational cost of updating the weights and bias for the l-th RBM in the p-th modality's pathway is $O\big(nk_1(s_p^{(l)}s_p^{(l+1)})\big)$. The cost is similar for other layers or for other pathway. For fine-tuning with k_2 epochs, the process is almost the same. Then the overall training complexity is:

$$O\big(n(k_1+k_2)\cdot\sum_{1\le p\le P}(\sum_{l=1}^{m_p}s_p^{(l)}s_p^{(l+1)}+m\cdot s_p^{(m_p+1)})\big).$$

Therefore, the training time complexity is linear to the size of the training data. These complexities guarantee the good scalability of our method.

5 EXPERIMENTS

In this section, we conduct experiments on real-world datasets to evaluate the performance of our method. We use image and text as multimodal data. And we define that the index of the image modality is 1 and the index of the text modality is 2. We first introduce the three real-world multimodal datasets. Then we introduce the experiment settings, evaluation metric and baselines we adopt. Finally we compare our method with other baseline and report the results. For the sake of simplicity, we use **DMHOR** (Deep Multimodal Hashing with Orthogonal Regularization) to represent our method.

5.1 Dataset

In our work, three real-world datasets are used for evaluation.

WIKI Dataset is a web document dataset, which has 2,866 documents from Wikipedia provided by [48]. Each document is accompanied by an image and is labelled by one of the ten semantic classes. Two documents are regarded as similar if they share the common class. The images are represented by 128-dimensional SIFT descriptors vectors, which are used as the image features. For text features, we use the probability distributions over 100 topics derived from a latent Dirichlet allocation (LDA) model [49] as the text feature vectors. 80% of the dataset is chosen for training set and the rest is for test set. The training set is used as the database set due to the limited samples in this dataset.

PASCAL [50] contains 1000 images which are randomly selected from 2008 PASCAL development kit. Each image is accompanied by five sentences, which can be used as the corresponded text data. These image-text pairs are labeled by 20 categories, each of which has 50 image-text pairs. These categories are used as the ground-truth. If two pairs share the common category, they are regarded to be similar. The images are represented by 512-dimensional GIST features. For text features, we use the probability distributions over 128 topics derived from a LDA model as the text feature vectors. 900 samples are chosen for training and the left 100 samples are for test. The training set is also adopted as the database set due to the limited samples in the dataset.

NUS-WIDE [51] is a web image dataset, which has 269,648 images from Flickr. These images are surrounded by some tags, with a total of 5,018 unique tags. In addition, ground-truth for 81 concepts in total is provided and each image is labelled by at least one concept. If two samples share at least one common concept, they are regarded to be similar in our experiment. We only consider images and their surrounding tags belonging to 10 largest concepts in our experiment. We randomly generate 3 sets of data. Each set contains 30,000 images as training set, 2,000 images as test set and 100,000 images as database set. For image features, We use 500-dimensional bag of words based on SIFT descriptors [52]. For text features, since some tags occur scarcely, we only consider the most 1,000 frequent tags and we use 1000-dimensional tag occurence vectors.

These three datasets not only fit our task of multimodal similarity search very much, but also have different properties, which can comprehensively prove the properties of the method. For example, the text modality of the three datasets are documents, sentences and tags. The size of the dataset ranges from 1k to 100k and the categories range from 10 to 20.

5.2 Experiment Settings

For the three datasets, our model consists of a 6-layer image pathway, a 4-layer text pathway and a joint RBM. The number of units in each layer is summarized in Table 2. The RBMs for the first layer are different and corresponded with the types of input. In NUS-WIDE, since the image input is real-valued and the text input is binary, the RBM for image input is Gaussian RBM and for text is Bernoulli RBM. In WIKI, the input of both modalities is real-valued, thus the RBMs for image and text input are both Gaussian RBMs. In PASCAL, since the image input is word-count vectors and the text input is real-valued, the RBM for image input is Replicated Softmax RBM and for text is Gaussian RBM. The RBMs for other layer are all Bernoulli RBMs.

TABLE 2: Number of Units in each layer

Dataset	Image Pathway	Text Pathway
WIKI	100-256-128-64-32-32	100-256-128-32
PASCAL	100-256-128-64-32-32	100-256-128-64
NUS-WIDE	500-512-256-128-64-32	1000-1024-512-128

We divide the whole dataset into small batches, each of which contains 100 samples. During the pretraining phase, each RBM is pretrained for 300 epochs. When updating the

weights and bias, the learning rate is 0.1, momentum starts at a value of 0.5 and is increased to 0.9 after 30 epochs. The weights are initialized by drawing from a normal distribution whose expectation is 0 and variance is 0.01. The bias terms are all initialized with zero vectors. When fine-tuning, we use gradient descents on each small batch for 200 epochs with three line searches performed. When performing binarization, we use the median value as the threshold to get the hash codes. The hyper-parameters of λ_1, λ_2 and μ are set as 0.5, 0.5 and 0.001 by using grid search. The value of $\alpha_{1,2}$, β_l^1 and β_l^2 are discussed later. We run experiments implemented by Matlab on a machine running Windows Server with 12 2.39GHz cores, 192 GB memory.

5.3 Evaluation Metrics

Our task focuses on the multi-source retrieval, which is defined in [36]. Given a query represented by multiple modalities in test set T, we will find the relevant items in the database D. Let $\Delta_i(j)$ be 1 if the i-th query and the j-th entity in database D are in the same class, otherwise be 0. $Dis(i,j)$ is the Hamming Distance between hash codes of the i-th item in T and the j-th item in D. Then after defining the threshold d, if the distance of two pairs is not larger than the threshold d, they are predicted to be similar by our model. Then we can compare our results with the ground-truth to evaluate the effectiveness of the method. In our experiment, we use *precision*, *recall* and *Mean Average Precision* (MAP) as evaluation metrics. The metrics can be defined as follows:

- *precision-recall* curve is a metric to evaluate the global performance of a method. The precision and recall within Hamming distance d are:

$$Pr(d) = \frac{\mid \{(i,j) \mid Dis(i,j) \leq d, i \in T, j \in D, \Delta_i(j) = 1\} \mid}{\mid \{(i,j) \mid Dis(i,j) \leq d, i \in T, j \in D\} \mid}$$
$$Re(d) = \frac{\mid \{(i,j) \mid Dis(i,j) \leq d, i \in T, j \in E, \Delta_i(j) = 1\} \mid}{\mid \{(i,j) \mid i \in T, j \in D, \Delta_i(j) = 1\} \mid}.$$

  Then we change the threshold distance d to record the corresponding *precision* and *recall* to get the *precision-recall* curve.
- *Mean Average Precision (MAP)* has good discrimination and stability and it is mainly a metric to evaluate the performance of ranking. It is calculated as follows:

$$AP(i) = \frac{\sum_j P@j(i) \cdot \Delta_i(j)}{\mid \{j \mid j \in D, \Delta_i(j) = 1\} \mid}$$
$$MAP = \frac{\sum_{i \in T} AP(i)}{\mid T \mid},$$

  where the $P@j(i)$ is defined as follows:
  Given a query, we sort the items in database according to the Hamming distance between the items and query. If some items have equal Hamming distance, they are sorted randomly. $index(i)$ is the index in the sorted database of the i-th item. $P@k$ for query i can be calculated as follows:

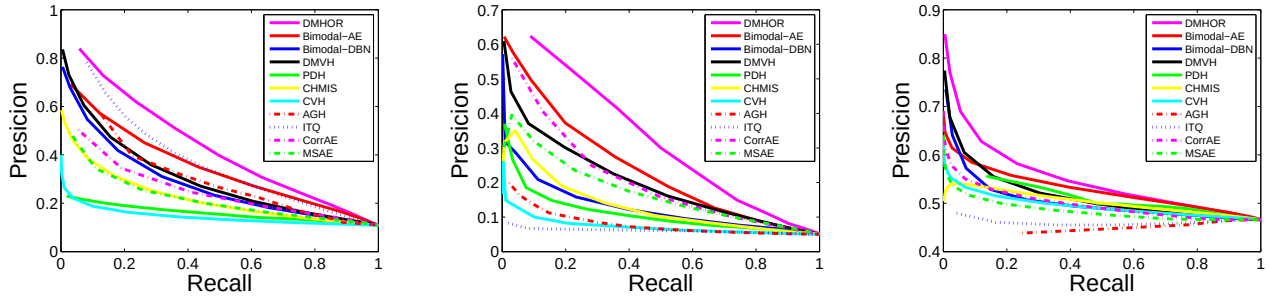
$$P@k(i) = \frac{\mid \{j \mid index(j) \leq k, j \in D, \Delta_i(j) = 1\} \mid}{k}.$$

5.4 Baseline Methods

To evaluate the effectiveness of the representations generated by our method, we adopt the following methods as baseline.

- Anchor-Graph Hashing (AGH) [53] proposes a graph-based hashing method to automatically capture the neighborhood structure inherent in the data to learn hash codes.
- Iterative Quantization Hashing (ITQ) [54] generates the hash function through minimizing the quantization error of mapping single modality data to the vertices of hypercube
- Cross View Hashing (CVH) [37] extends spectral hashing to multiview to learn the hash function to do cross-view similarity search.
- Predictable Dual-View Hashing (PDH) [38] represents the distribution of samples by non-orthogonal bases and employs a max-margin formulation to learn hash function to do cross-modality similarity search.
- Composite Hashing with Multiple Information Sources (CHMIS) [31] integrates multiple information sources to get hash codes by adjusting weights for each individual modalities to get a better performance.
- Deep Multiview Hashing (DMVH) [15] proposes a deep network, where each layer contains view-specific and shared hidden units, to do multimodal hashing.
- BIMODAL-DBN [6] proposes a multimodal Deep Belief Network to connect multiple modalities to learn representation for multimodal data.
- BIMODAL-AE [12] proposes a multimodal Deep Autoencoder incorporating with a cross-modality autoencoder to learn multimodal representations.
- Correspondence Autoencoder (CORR-AE) [41] proposes a correspondence autoencoder to correlate hidden representations of two uni-modal autoencoders to solve the problem of cross-modal retrieval.
- Multimodal Stacked Autoencoder (MSAE) [55] uses stacked autoencoder to map multimodal objects into a common space to do cross-modality retrieval.

Note that AGH and ITQ are hashing for single modality input. Thus, we concatenate data of multiple modalities as the input to make a fair comparison. CVH, PDH and CHMIS are multimodal hashing using shallow models. They are adopted as baselines because comparing with them can comprehensively prove the effectiveness of our deep models under the hashing scenario. BIMODAL-DBN, BIMODAL-AE, CORR-AE and MSAE are deep models for multimodal data. But they are not specially designed for generating compact hash codes. We binarize the representations generated by them to make binary hash codes. Note that DMVH is multimodal hashing using deep models. It is a supervised method in fine-tuning. Therefore, we apply multimodal autoencoder in this phase to make a fair comparison. In addition, all the deep models adopt dropout to make fair comparisons.



(a) WIKI (b) PASCAL (c) NUS-WIDE

Fig. 3: Precision-recall curve for WIKI, PASCAL and NUS-WIDE dataset, with a fixed bit number of 16.

5.5 Results

In this section, we compare the performance of hash codes generated by different algorithms when given both image and corresponding text information during training and testing. We run each algorithms five times and report the following average results.

First of all, in all datasets we vary the the number of hash bits in $\{8, 16, 32, 64, 128\}$ to see the MAP of different algorithms on three datasets, as the Table 3 shows.

From these comparison results, some observations and analysis are included as follows:

- In most cases, the deep-structured models can consistently and obviously outperform other shallow-structured models, which demonstrates that the multimodal correlations cannot be well captured by shallow-structured models, and deep models have much merit in this aspect due to their intrinsic nonlinearity.
- When the length of codes increases, the performance of our method improves much more than that of other baseline methods. For example, when length increases from 8 to 128, MAP of **DMHOR** on WIKI increases almost by 0.2, which is far more than other baselines. The reason is that our methods well reduce the redundant information, thus it can make use of the increasing bits to represent useful information.
- The result shows that DMHOR outperforms other methods more in WIKI and Pascal than the improvements in NUS-WIDE. The reasons are manifold. Some reasons can be that WIKI provides articles and Pascal provides sentences, which can provide much more information than that of NUS-WIDE, which only provides tags. Other reasons can be that the image quality in WIKI and Pascal is better, and DMHOR can extract more visual information in high quality visual data.

By fixing the length of hash codes to 16, we further report the precision-recall curve. The curve on WIKI dataset is shown in Figure 3(a), on PASCAL is shown in Figure 3(b) and on NUS-WIDE dataset is shown in Figure 3(c). From the figure, it is clear that **DMHOR** shows the best performance.

Performing binarization on real-valued representations to generate binary hash codes will loss some information.

TABLE 3: MAP on three datasets with varing length of hash codes

(a) WIKI

Algorithm	8 bit	16 bit	32 bit	64 bit	128 bit
DMHOR	**0.3424**	**0.489**	**0.5268**	**0.5281**	**0.5304**
BIMODAL-AE	0.306	0.4122	0.4478	0.4423	0.4221
BIMODAL-DBN	0.2695	0.3727	0.4035	0.4087	0.4067
DMVH	0.2847	0.3543	0.3960	0.3915	0.3824
CORR-AE	0.2321	0.3073	0.3135	0.2848	0.2525
MSAE	0.2216	0.2837	0.3037	0.3118	0.2890
CHMIS	0.2171	0.2672	0.2697	0.2474	0.2293
CVH	0.17	0.1649	0.1824	0.1405	0.1358
PDH	0.1618	0.1602	0.2532	0.1829	0.1837
ITQ	0.3042	0.418	0.4147	0.3930	0.3927
AGH	0.3069	0.3777	0.36	0.3148	0.2866

(b) Pascal

Algorithm	8 bit	16 bit	32 bit	64 bit	128 bit
DMHOR	**0.2595**	**0.3919**	**0.4694**	**0.4632**	**0.4677**
BIMODAL-AE	0.2305	0.2852	0.3219	0.3333	0.3324
BIMODAL-DBN	0.1606	0.1872	0.223	0.2702	0.2669
DMVH	0.1580	0.2515	0.3203	0.3365	0.3291
CORR-AE	0.1569	0.2159	0.2508	0.2585	0.2602
MSAE	0.1490	0.1726	0.2206	0.2578	0.2312
CHMIS	0.1418	0.1744	0.1712	0.1351	0.1233
CVH	0.0818	0.0912	0.1165	0.1435	0.1426
PDH	0.1134	0.1283	0.1575	0.1766	0.1838
ITQ	0.0857	0.0871	0.1232	0.1498	0.1594
AGH	0.0975	0.1271	0.1405	0.1377	0.1351

(c) NUS-WIDE

Algorithm	8 bit	16 bit	32 bit	64 bit	128 bit
DMHOR	**0.5472**	**0.5618**	**0.5791**	**0.5809**	**0.5805**
BIMODAL-AE	0.5314	0.5476	0.5508	0.5511	0.5482
BIMODAL-DBN	0.5252	0.5392	0.5386	0.5409	0.5392
DMVH	0.5211	0.5288	0.5336	0.5351	0.5355
CORR-AE	0.5023	0.5078	0.5169	0.5160	0.5118
MSAE	0.5001	0.5081	0.5113	0.5092	0.5077
CHMIS	0.5199	0.5288	0.5294	0.5009	0.5016
CVH	0.5088	0.4961	0.4868	0.4789	0.4769
PDH	0.5164	0.5169	0.5223	0.5247	0.5242
ITQ	0.4964	0.4917	0.4953	0.4959	0.4980
AGH	0.495	0.4931	0.4894	0.4866	0.4840

Therefore, to comprehensively prove the binary representations generated by our methods are effective, we also need to compare them with real-valued representations generated by baseline methods. Since BIMODAL-AE performs best among all the baseline methods, we compare the binary representations generated by our hash method with the real-

TABLE 4: MAP and Time comparisons for DMHOR and Real-Bimodal-AE on WIKI dataset. The value of the time includes the time cost for retrieving the overall 693 queries. The training time unit is millisecond. The test time unit is second

Method	MAP					TRAINING TIME (s) / QUERY TIME (ms)				
	8 bit	16 bit	32 bit	64 bit	128 bit	8 bit	16 bit	32 bit	64 bit	128 bit
DMHOR	**0.3215**	**0.4732**	**0.5171**	**0.5232**	**0.5251**	**209.8/464**	**210.1/487**	**218.4/533**	**221.9/563**	**229.9/675**
REAL-BIMODAL-AE	0.2642	0.3937	0.4716	0.4729	0.4731	325.41/3068	327.6/3450	334.1/4080	340.13/5571	350.4/6016

valued representations generated by BIMODAL-AE, denoted as REAL-BIMODAL-AE. Besides, considering that Bimodal-AE does not adopt dropout, we do not use dropout in DMHOR to fairly compare the representations generated by them. In addition, we also report the training and retrieval efficiency of **DMHOR** and REAL-BIMODAL-AE. The results are shown in Table 4. From the results, we can get the following observations and analysis.

- Although our representations perform binarization, our method still gains a substantial improvements over real-valued representations learned by BIMODAL-AE. Thus, the representations learned by our methods are effective.
- In terms of the time cost of training, our method spends less time. Furthermore, for the time cost of online retrieval, our method takes nearly one tenth of the time to perform retrieval for a query compared with the real-valued representations learned by REAL-BIMODAL-AE. The reason for it is that the representations of our method are binary.
- From the results, we can use only 16-bit binary representations of **DMHOR** to achieve the performance of 128-bit real-valued representations of REAL-BIMODAL-AE. In this way, we can save a lot of time and space in real-world applications but still get satisfied performance.

Therefore, even though comparing with real-valued representations, the binary representations generated by our method are still more effective and far more efficient.

Finally, we show some concrete examples on three datasets to compare our **DMHOR** method with the best baseline method BIMODAL-AE. For the same query, we report the top five retrieved results of **DMHOR** and BIMODAL-AE. The result is shown in Figure 4. From the result, it is obvious that our method performs better than the baseline methods. Furthermore, even though some images retrieved by our method and baseline methods are relevant to the query, it is intuitive that images retrieved by our method are more relevant. For example, as shown in Figure 4(c), the query is belonging to the building category and the scene of the query image is in the evening. All the results of our method depict the buildings at nightfall or in the evening. But the third image retrieved by the BIMODAL-AE is the buildings in the daytime. Thus, our method can extract more relevant factors and return highly relevant retrieved results for the retrieval task.

5.6 Observations

Here we will give some insights about our proposed methods. Firstly, we will evaluate how the orthogonality constraints we proposed affects the performance. In Eq. 10, we fix the values of α_l^2 to be 0.5 from $l = 1$ to 4 and change the value of α_l^1 and $\gamma_{p,q}$ to observe the change of MAP for WIKI. All of the parameters are optimally chosen.

In Table 5, Exp1 (Exp10) and Exp9 (Exp 18) impose orthogonality constraints on all the weight matrices as shown in Eq. 8, but Exp9 (Exp18) does not impose cross-modality constraints as shown in Eq. 9. From Exp2 to Exp8, we gradually remove the orthogonality constraints from low layer to high layer. From Exp11 to Exp17, we gradually remove the orthogonality constraints from high layer to low layer. From the results of Table 5, we can make the following observations and anaylsis:

- The result that Exp1 outperforms Exp2 to Exp8 and Exp10 outperforms Exp11 to Exp17 demonstrates the effectiveness of the orthogonality constraints on each layer's matrix.
- The result that Exp1 outperforms Exp9 demonstrates that the cross-modality constraint is necessary and effective.
- The result shows that from Exp1 to Exp8 the performance decreases more and more and from Exp10 to Exp17 the performance decreases less and less. It indicates that the orthogonality constraint of the higher layer is more important to the improvements of performance than that of lower layer. The reason for it can be that the higher layer has less number of bits. Thus, reducing the redundant information of the higher layer will be more valuable.

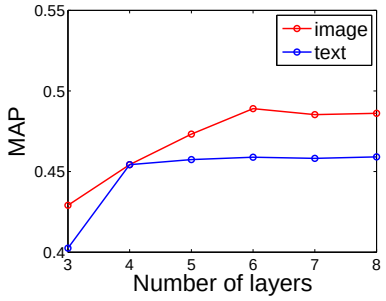


Fig. 5: MAP on WIKI by fixing the number of layers of one modality to 4 and varing the number of layers of another modality. The result shows text modality becomes top earlier than image modality.

Now we evaluate how the number of layers affects the performance. We change the number of layers of one modality and set the number of another modality to 4 layers. The result is shown in Figure 5. From the results, we find that the curve of text modality stabilizes faster than that of image modality. The explanation is that image features

TABLE 5: MAP on WIKI with varying orthogonality constraints with 16-bit codes

	α_6^1	α_5^1	α_4^1	α_3^1	α_2^1	α_1^1	$\beta_{1,2}$	MAP		α_6^1	α_5^1	α_4^1	α_3^1	α_2^1	α_1^1	$\beta_{1,2}$	MAP
Exp1	0.5	0.5	0.5	0.5	2	5	0.5	**0.489**	Exp10	0.5	0.5	0.5	0.5	2	5	0.5	**0.489**
Exp2	0.5	0.5	0.5	0.5	2	0	0.5	0.485	Exp11	0	0.5	0.5	0.5	2	5	0.5	0.459
Exp3	0.5	0.5	0.5	0.5	0	0	0.5	0.478	Exp12	0	0	0.5	0.5	2	5	0.5	0.436
Exp4	0.5	0.5	0.5	0	0	0	0.5	0.467	Exp13	0	0	0	0.5	2	5	0.5	0.415
Exp5	0.5	0.5	0	0	0	0	0.5	0.45	Exp14	0	0	0	0	2	5	0.5	0.402
Exp6	0.5	0	0	0	0	0	0.5	0.429	Exp15	0	0	0	0	0	5	0.5	0.396
Exp7	0	0	0	0	0	0	0.5	0.391	Exp16	0	0	0	0	0	0	0.5	0.391
Exp8	0	0	0	0	0	0	0	0.378	Exp17	0	0	0	0	0	0	0	0.378
Exp9	0.5	0.5	0.5	0.5	2	5	0	0.475	Exp18	0.5	0.5	0.5	0.5	2	5	0	0.475

have larger semantic gap, thus we need to assign more layers to attain a better performance. However, for text modality, the structure needs not to be very deep otherwise it will waste time and space to train the structure but obtain almost the same performance. Therefore, we need to assign an appropriate number of layers for different modalities. 4 layers for text modality and 6 layers for image modality is optimal for us.

6 CONCLUSIONS

In this paper, we propose a hashing-based orthogonal deep model to learn compact binary codes for multimodal representations. The model can well capture the intra-modality and inter-modality correlation to make the codes accurate. Moreover, the proposed orthogonal regularized deep structure solves the redundancy problem, which makes the representation compact. We optimize the compactness and discriminability together in a unified model to make a good trade-off between them in the learned representations. Furthermore, the proposed modality-specific structure of applying different numbers of layers to different modalities makes an effective representation and compact learning process. Experimental results demonstrate a substantial gain of our method compared with baseline methods on three widely used public datasets.

For future work, we will aim to design intelligent strategy to automatically determining the ideal number of layers for different modalities. In this case, it will save a lot of time for humans to find the optimal number of layers for different modalities.

REFERENCES

[1] X. Li, C. Shen, A. Dick, and A. van den Hengel, "Learning compact binary codes for visual tracking," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 2419–2426.

[2] H. Fan, M. Yang, Z. Cao, Y. Jiang, and Q. Yin, "Learning compact face representation: Packing a face into an int32," in *Proceedings of the ACM International Conference on Multimedia*. ACM, 2014, pp. 933–936.

[3] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[4] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural networks*, vol. 13, no. 4, pp. 411–430, 2000.

[5] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1, pp. 37–52, 1987.

[6] N. Srivastava and R. Salakhutdinov, "Learning representations for multimodal data with deep belief nets," in *International Conference on Machine Learning Workshop*, 2012.

[7] Y. Bengio, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[8] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks." in *NIPS*, vol. 1, no. 2, 2012, p. 4.

[10] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "Joint learning of words and meaning representations for open-text semantic parsing," in *International Conference on Artificial Intelligence and Statistics*, 2012, pp. 127–135.

[11] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *JMLR*, vol. 11, pp. 625–660, 2010.

[12] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *ICML*, 2011, pp. 689–696.

[13] N. Srivastava and R. Salakhutdinov, "Multimodal learning with deep boltzmann machines." in *NIPS*, 2012, pp. 2231–2239.

[14] M. Norouzi and D. M. Blei, "Minimal loss hashing for compact binary codes," in *ICML*, 2011, pp. 353–360.

[15] Y. Kang, S. Kim, and S. Choi, "Deep learning to hash with multiple representations." in *ICDM*, 2012, pp. 930–935.

[16] J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, Q. V. Le, and A. Y. Ng, "On optimization methods for deep learning," in *ICML*, 2011, pp. 265–272.

[17] M. Guillaumin, J. Verbeek, and C. Schmid, "Multimodal semi-supervised learning for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 902–909.

[18] M. J. Huiskes, B. Thomee, and M. S. Lew, "New trends and ideas in visual concept detection: the mir flickr retrieval evaluation initiative," in *Proceedings of the international conference on Multimedia information retrieval*. ACM, 2010, pp. 527–536.

[19] S. Roller and S. S. Im Walde, "A multimodal lda model integrating textual, cognitive and visual modalities," *Seattle, Washington, USA*, pp. 1146–1157, 2013.

[20] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *SOCG*. ACM, 2004, pp. 253–262.

[21] Q. Lv, W. Josephson, Z. Wang, M. Charikar, and K. Li, "Multi-probe lsh: efficient indexing for high-dimensional similarity search," in *VLDB*. VLDB Endowment, 2007, pp. 950–961.

[22] A. Joly and O. Buisson, "A posteriori multi-probe locality sensitive hashing," in *ACM MM*. ACM, 2008, pp. 209–218.

[23] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing." in *NIPS*, vol. 9, no. 1, 2008, p. 6.

[24] R. Salakhutdinov and G. Hinton, "Semantic hashing," *IJAR*, vol. 50, no. 7, pp. 969–978, 2009.

[25] A. Torralba, R. Fergus, and Y. Weiss, "Small codes and large image databases for recognition," in *CVPR*. IEEE, 2008, pp. 1–8.

[26] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, "Supervised hashing for image retrieval via image representation learning," in *AAAI*, 2014.

[27] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang, "Supervised hashing with kernels," in *CVPR*. IEEE, 2012, pp. 2074–2081.

[28] M. Norouzi, D. J. Fleet, and R. Salakhutdinov, "Hamming distance metric learning." in *NIPS*, 2012, pp. 1070–1078.

[29] J. Wang, S. Kumar, and S.-F. Chang, "Semi-supervised hashing for scalable image retrieval," in *CVPR*. IEEE, 2010, pp. 3424–3431.

[30] M. M. Bronstein, A. M. Bronstein, F. Michel, and N. Paragios, "Data fusion through cross-modality metric learning using similarity-sensitive hashing," in *CVPR*. IEEE, 2010, pp. 3594–3601.

[31] D. Zhang, F. Wang, and L. Si, "Composite hashing with multiple information sources," in *SIGIR*. ACM, 2011, pp. 225–234.

[32] Y. Zhen and D.-Y. Yeung, "A probabilistic model for multimodal hash function learning," in *SIGKDD*. ACM, 2012, pp. 940–948.

[33] D. Zhai, H. Chang, Y. Zhen, X. Liu, X. Chen, and W. Gao, "Parametric local multimodal hashing for cross-view similarity search," in *IJCAI*. AAAI Press, 2013, pp. 2754–2760.

[34] J. Song, Y. Yang, Y. Yang, Z. Huang, and H. T. Shen, "Inter-media hashing for large-scale retrieval from heterogeneous data sources," in *SIGMOD*. ACM, 2013, pp. 785–796.

[35] F. Wu, Z. Yu, Y. Yang, S. Tang, Y. Zhang, and Y. Zhuang, "Sparse multi-modal hashing," *Multimedia, IEEE Transactions on*, vol. 16, no. 2, pp. 427–439, 2014.

[36] D. Zhang and W.-J. Li, "Large-scale supervised multimodal hashing with semantic correlation maximization," in *AAAI*, 2014.

[37] S. Kumar and R. Udupa, "Learning hash functions for cross-view similarity search," in *IJCAI*, vol. 22, no. 1, 2011, p. 1360.

[38] M. Rastegari, J. Choi, S. Fakhraei, D. Hal, and L. Davis, "Predictable dual-view hashing," in *ICML*, 2013, pp. 1328–1336.

[39] M. Ou, P. Cui, F. Wang, J. Wang, W. Zhu, and S. Yang, "Comparing apples to oranges: a scalable solution with heterogeneous hashing," in *SIGKDD*. ACM, 2013, pp. 230–238.

[40] W. Wang, B. C. Ooi, X. Yang, D. Zhang, and Y. Zhuang, "Effective multi-modal retrieval based on stacked auto-encoders," *Proceedings of the VLDB Endowment*, vol. 7, no. 8, 2014.

[41] F. Feng, X. Wang, and R. Li, "Cross-modal retrieval with correspondence autoencoder," in *ACM MM*. ACM, 2014, pp. 7–16.

[42] G. Hinton, "A practical guide to training restricted boltzmann machines," *Momentum*, vol. 9, no. 1, p. 926, 2010.

[43] M. Welling, M. Rosen-Zvi, and G. E. Hinton, "Exponential family harmoniums with an application to information retrieval," in *NIPS*, 2004, pp. 1481–1488.

[44] G. E. Hinton and R. Salakhutdinov, "Replicated softmax: an undirected topic model," in *NIPS*, 2009, pp. 1607–1614.

[45] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[46] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[47] Z. Wu, Y.-G. Jiang, J. Wang, J. Pu, and X. Xue, "Exploring inter-feature and inter-class relationships with deep neural networks for video classification," in *Proceedings of the ACM International Conference on Multimedia*. ACM, 2014, pp. 167–176.

[48] 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*. ACM, 2010, pp. 251–260.

[49] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *JMLR*, vol. 3, pp. 993–1022, 2003.

[50] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth, "Every picture tells a story: Generating sentences from images," in *Computer Vision–ECCV 2010*. Springer, 2010, pp. 15–29.

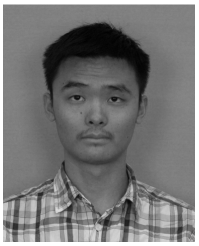
[51] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "Nus-wide: a real-world web image database from national university of singapore," in *CIVR*. ACM, 2009, p. 48.

[52] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. Ieee, 1999, pp. 1150–1157.

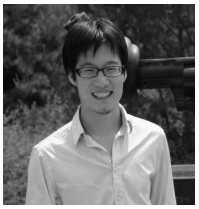
[53] W. Liu, J. Wang, S. Kumar, and S.-F. Chang, "Hashing with graphs," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 1–8.

[54] Y. Gong and S. Lazebnik, "Iterative quantization: A procrustean approach to learning binary codes," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 817–824.

[55] W. Wang, B. C. Ooi, X. Yang, D. Zhang, and Y. Zhuang, "Effective multi-modal retrieval based on stacked auto-encoders," *PVLDB*, pp. 649–660, 2014.



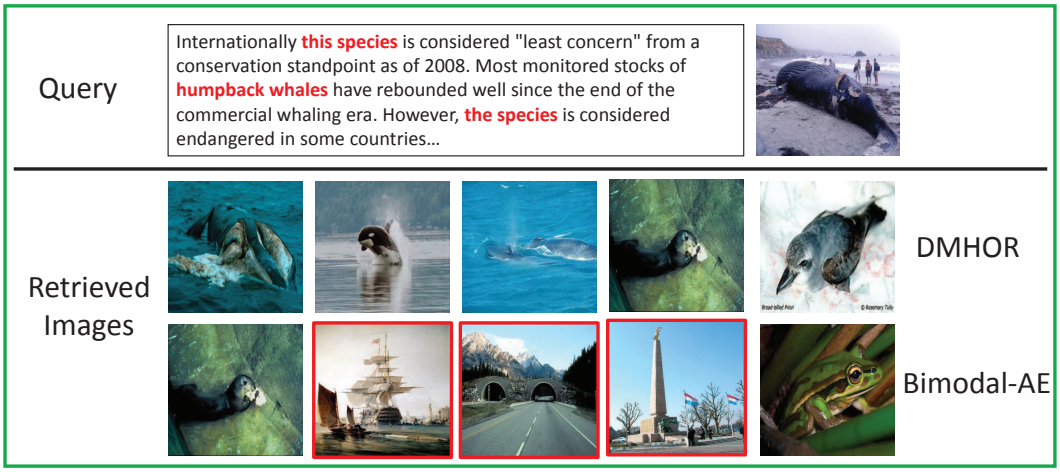
Daixin Wang received the B.E. degree from the Department of Computer Science and Technology of Tsinghua University in 2013. He is a second-year Ph.D. candidate in the Department of Computer Science and Technology of Tsinghua University. His main research interests include multimedia, image retrieval and deep learning.



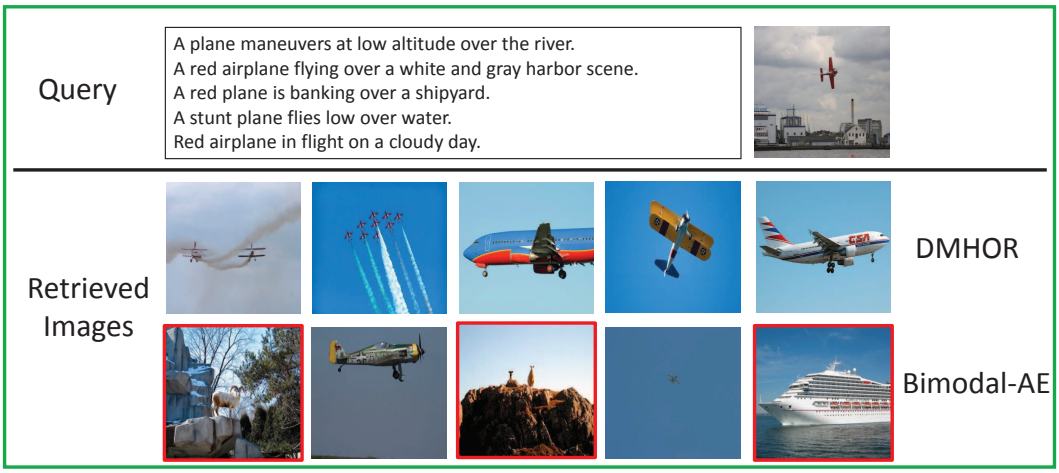
Peng Cui received the Ph.D. degree in computer science in 2010 from Tsinghua University and he is an Assistant Professor at Tsinghua. He has vast research interests in data mining, multimedia processing, and social network analysis. Until now, he has published more than 20 papers in conferences such as SIGIR, AAAI, ICDM, etc. and journals such as IEEE TMM, IEEE TIP, DMKD, etc. Now his research is sponsored by National Science Foundation of China, Samsung, Tencent, etc. He also serves as Guest Editor, Co-Chair, PC member, and Reviewer of several high-level international conferences, workshops, and journals.

Mingdong Ou received the BSc degree in computer science from Tsinghua University. He is currently a PhD candidate in computer science at Tsinghua University. His work focus on developing scalable representation learning methods for search and recommendation on massive heterogeneous data.

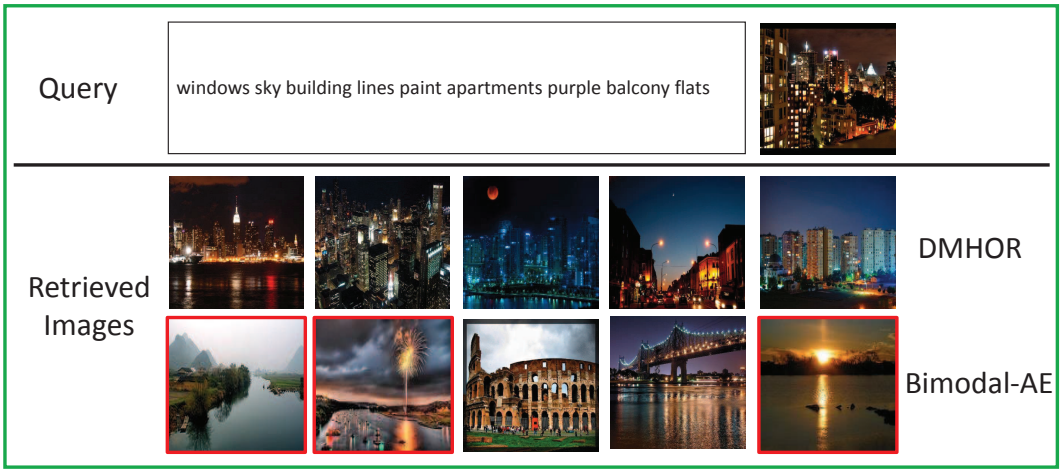
Wenwu Zhu received the Ph.D. degree from Polytechnic Institute of New York University in 1996. He is now a Professor at Tsinghua University. His research interest is wireless/Internet multimedia communication and computing. He worked at Bell Labs during 1996 to 1999. He was with Microsoft Research Asia's Internet Media Group and Wireless and Networking Group as research manager from 1999 to 2004. He was the director and chief scientist at Intel Communication Technology Lab, China. He was also a senior researcher at the Internet Media Group at Microsoft Research Asia. He has published more than 200 referred papers and led 40 patents. He participated in the IETF ROHC WG on robust TCP/IP header compression over wireless links and IEEE 802.16m WG standardization. He currently serves as Chairman of IEEE Circuits and System Society Beijing Chapter, advisory board on the International Journal of Handheld Computing Research, and President of ACM Beijing.



(a) WIKI



(b) PASCAL



(c) NUS-WIDE

Fig. 4: Concrete examples on three datasets using our **DMHOR** method and the best baseline method BIMODAL-AE. In each dataset, the query image and its corresponding text description are shown in the first line. Retrieved images of **DMHOR** are shown in the second line. Retrieved images of the baseline BIMODAL-AE are shown in the last line. Irrelevant images with the query are with red bounding box. The categories in WIKI, PASCAL and NUS-WIDE are biology, aeroplane and buildings respectively.