# KERNEL-BASED SUPERVISED HASHING FOR CROSS-VIEW SIMILARITY SEARCH

*Jile Zhou, Guiguang Ding, Yuchen Guo, Qiang Liu, XinPeng Dong*

School of Software, Tsinghua University, Beijing, P.R.China
jile.zip@gmail.com, dinggg@tsinghua.edu.cn, yuchen.w.guo@gmail.com,
liuqiang@mail.tsinghua.edu.cn dongxinpeng_888@163.com

## ABSTRACT

Spectral-based hashing (SpH) is the most used method for cross-view hash function learning (CVHFL). However, the following three problems are shared by many existing SpH methods. Firstly, preserving intra- and inter-similarity simultaneously increases models' complexity significantly. Secondly, linear model applied in many SpH methods is hard to handle multimodal data in cross-view scenarios. Thirdly, to learn irrelevant multiple bits, SpH imposes orthogonality constraints which decreases the mapping quality substantially with the increase of bit number. To address these challenges, we propose a novel SpH method for CVHFL in this paper, referred to as Kernel-based Supervised Hashing for Cross-view Similarity Search (KSH-CV). We prove that the intra-adjacency matrix is redundant given inter-adjacency matrix. Then we define our objective function in a supervised and kernelized way which just needs to preserve inter-similarity. Furthermore a novel Adaboost algorithm, which minimizes exponential mapping loss function for cross-view similarity search, is derived to solve the objective function efficiently while avoiding orthogonality constraints. Extensive experiments verifies that KSH-CV can significantly outperform several state-of-the-art methods on three cross-view datasets.

## 1. INTRODUCTION

Similarity search, a method of searching semantically related results from database for a given query, lays the foundation for many important applications, such as information retrieval and object detection. One traditional method of similarity search is scanning over databases to find the nearest neighbors, which is quite inefficient and expensive to compute similarity among floating/integer descriptors. Whereas, with the explosive growth of multimedia data on the Internet, similarity search systems face efficiency challenges in large datasets. A common solution is to employ the sub-linear tree-based method, but it only works on low-dimension data [1].

Hashing-based methods, which embed high-dimension data into compact binary codewords, is quite efficient in both storage and time for similarity search. One of the most famous hashing-based models is locality-sensitive hashing (LSH) [1], whose idea is to map data from original space to Hamming space while preserving their similarity with high probability. As extensions of standard LSH, some machine learning techniques are employed to design effective compact hashing, such as Manifold Learning, Supervised Learning, Kernel Learning, Quantization Learning, K-means and PCA, which respectively generate Spectral Hashing [2], Supervised Hashing [3], Kernelized Hashing [4], Iterative Quantization Hashing [5], K-means Hashing [6] and PCA Hashing [7].

However, the bulk of LSH methods above are single-view while many real-world applications are cross-view. The task of similarity search on cross-view datasets is to retrieve similar results of all views for a given query. Taking Wikipedia as example. Pages may contain images, texts, or both. When a query word or picture is given, the system should return both relevant articles and images. Another example is CBIR. An image can be represented by many visual features, e.g. SIFT and CEDD [8], and a user can use any kind of descriptor as the query to retrieve on databases with several features.

Recently, a few cross-view methods have shown up, and SpH is quite widely-used, e.g. [9, 10, 11, 12]. SpH learns compact binary codes by minimizing the average weighted distance of codewords. However, most existing SpH for cross-view share three shortcomings. Firstly, preserving inter-similarity and intra-similarity simultaneously increases the complexity of models. Secondly, linear model applied in SpH methods is hard to handle multimodal data sampled from different probability distributions for CVHFL. Thirdly, SpH imposes orthogonality constraints to approximately decorrelate the hash bits. However the constraints decrease the mapping quality substantially when increasing bit number, since most of the variance is contained in top few eigenvectors [7].

Inspired by supervised learning and kernel learning, we propose a novel *Kernel-based Supervised Hashing for Cross-view Similarity Search* (KSH-CV) method for CVHFL. Our paper makes several contributions for CVHFL as follows:

1. We analysis theoretical property of locally-sensitive preserving in CVHFL, and we prove that the intra-adjacency matrix is redundant, because it can be approached by inter-adjacency matrix. Hence, KSH-CV only takes inter-similarity into consideration, which

can reduce the complexity of models significantly.

2. We observed that preserving inter-similarity for cross-view is essentially equivalent to classification problem, hence kernelized model shows much better performance than linear model on multimodal datasets. And we are the first to use kernelized model in CVHFL.

3. In order to avoid orthogonality constraints of SpH , we utilize Adaboost algorithm to replace eigenvalue-decomposition. Different from the standard Adaboost algorithm, which minimizes exponential classification error, we derive a novel Adaboost algorithm based on exponential mapping loss function for CVHFL, to measure the mapping quality for inter-similarity preserving.

Experiments on three datasets show that KSH-CV can significantly outperform several sate-of-the-art methods, verifying the effectiveness of KSH-CV under cross-view scenarios.

## 2. RELATED WORK

Efficiency is critical for similarity search in many applications, especially in large databases with high-dimension data, where traditional methods tend to break down. *Approximate similarity search* is proved as a good tradeoff between accuracy and computational complexity, and LSH is the most notable method. A lot of work has been done on LSH for single-view [1, 2, 3, 4], and cross-view similarity search has drawn more and more attention recently [9, 10, 11, 13, 14, 12].

SpH methods for cross-view, whereas, not only consider intra-similarity between datapoints for single-view, but also focus on inter-similarity over different views. Suppose that $\mathcal{O} = \{o_i\}_{i=1}^n$ is a set of multi-view objects, where $\mathbf{x}_i^{(t)}$ and $\mathbf{y}_i^{(t)}$ are the $t$-th view and the $t$-th codewords of object $o_i$. The objective functions for most SpH methods for cross-view share the same formulation written as follows:

$$\begin{aligned} &\text{minimize} \sum_{t,t'} \sum_{ij} W_{ij}^{(t,t')} d_{ij}^{(t,t')} \\ &s.t. \text{some constraint} \end{aligned} \tag{1}$$

where $d_{ij}^{(t,t')}$ is the distance between $\mathbf{y}_i^{(t)}$ and $\mathbf{y}_j^{(t')}$, and $W_{ij}^{(t,t')}$ is the similarity between $\mathbf{x}_i^{(t)}$ and $\mathbf{x}_j^{(t')}$ representing intra-similarity when $t = t'$ and inter-similarity otherwise.

CHMIS [9] learns a binary representation $\mathbf{y}_i^*$ to represent an object $o_i$ for all views. The distance is defined as $d_{ij}^{(t,t')} = ||\mathbf{y}_i^* - \mathbf{y}_j^*||^2$, and the elements of the average similarity matrix is defined as $W_{ij} = \sum_t W_{ij}^{(t,t)}$. Similar to CHMIS, MVSH [11] learns an integrated binary code $\mathbf{y}_i^*$, and constructs an average similarity matrix $W$. To avoid undesirable embedding, MVSH defines distance measure as $d_{ij}^{(t,t')} = (\mathbf{y}_i^*)^T \mathbf{y}_j^*$. CVH [10] designs a set of binary codes $\{\mathbf{y}_i^{(t)}\}$ for each view of $o_i$, and minimises the cumulative distance $d_{ij}^{(t,t')} = \sum_t \sum_{t' \geq t} \frac{1}{4} ||\mathbf{y}_i^{(t)} - \mathbf{y}_j^{(t')}||$. Inter-media Hashing (IMH) [12] uses formula (1) to constrain hash codes, then uses linear regression with regularization model to learn hash functions simultaneously. However, CHMIS and MVSH may be inappropriate for real-word cross-view similarity search tasks, because they are able to generate codeword only when all views of a query are given. Furthermore, CVH and IMH share the same shortcomings as SpH model mentioned above.

## 3. KERNEL-BASED SUPERVISED HASHING FOR CROSS-VIEW

In this Section, we will show how to learn hash functions in bimodal case as it is the most important scene in real world.

### 3.1. Problem Formulation

Suppose that $\mathcal{O} = \{o_i\}_{i=1}^n$ is a set of multi-view objects and $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$, $\mathcal{Y} = \{\mathbf{y}_i\}_{i=1}^n$ are two different views of $\mathcal{O}$, where $\mathbf{x}_i \in \mathbf{R}^{d_\mathcal{X}}$ and $\mathbf{y}_i \in \mathbf{R}^{d_\mathcal{Y}}$(usually, $d_\mathcal{X} \neq d_\mathcal{Y}$). The purpose of CVHFL is to learn two groups of hash functions: $H_\mathcal{X} = [h_\mathcal{X}^{(1)}, ..., h_\mathcal{X}^{(k)}] : \mathbf{R}^{d_\mathcal{X}} \mapsto \{1, -1\}^k$, $H_\mathcal{Y} = [h_\mathcal{Y}^{(1)}, ..., h_\mathcal{Y}^{(k)}] : \mathbf{R}^{d_\mathcal{Y}} \mapsto \{1, -1\}^k$, which preserve intra- and inter-similarity efficiently, and then the object $o_i$ is projected as $k$-dimension binary bits in two views respectively.

### 3.2. Preserving Similarity for Cross-View

Different from single-view hashing function learning with intra-similarity, CVHFL involves inter-similarity among objects. We define two projection types for CVHFL as follows:

**Definition 1.** *The projection is intra-similarity preserved, if the similar objects in the same view are mapped to the same bin. The projection is inter-similarity preserved, if the different views of the similar object are mapped to the same bin.*

Obviously, an effictive algorithm for CVHFL should preserve intra- and inter-similarity simultaneously. But integrating both constraints into objective function increases the complexity of model. Fortunately, under the framework of bipartite graph [15], we can prove that the intra-adjacency matrix is redundant when inter-adjacency matrix is given.

Let $W^{inter}$ be the inter-adjacency matrix between image and text, i.e. $W_{pq}^{inter}$ denotes the inter-similarity between image $i_p$ and text $t_q$. To explicitly capture the image-to-text relationship, we introduce a bipartite graph $\mathfrak{B}(\mathcal{I}, \mathcal{T}, \mathcal{E})$. The node sets $\mathcal{I} = \{i_p\}_{p=1}^n, \mathcal{T} = \{t_p\}_{p=1}^n$ represents image points and text points respectively, and $\mathcal{E}$ contains edges connecting $\mathcal{I}$ and $\mathcal{T}$. We connect an undirected edge between $i_p$ and $t_q$ if and only if $W_{pq}^{inter} > 0$. Based on $\mathfrak{B}(\mathcal{I}, \mathcal{T}, \mathcal{E})$, the 2-order intra-similarity can be obtained by the transition probabilities in two time step, i.e. $W_{pq}^{intra} = p^{(2)}(i_p|i_q) = p^{(2)}(t_p|t_q)$.

**Theorem 1.** *The 2-order intra-similarity can be deduced by inter-similarity based on the chain rule of Markov random walks on* $\mathfrak{B}(\mathcal{I}, \mathcal{T}, \mathcal{E})$.

*Proof.* The transition probabilities in one time step can be obtained as follows:

$$p^{(1)}(i_p|t_q) = \frac{W_{pq}^{inter}}{\sum_k W_{kq}^{inter}}, p^{(1)}(t_p|i_q) = \frac{W_{pq}^{inter}}{\sum_k W_{kq}^{inter}} \quad (2)$$

Exploiting the chain rule of Markov random walks, we have:

$$\begin{aligned} p^{(2)}(i_p|i_q) &= \sum_s p^{(1)}(i_p|t_s)p^{(1)}(t_s|i_q) \\ &= \sum_s \frac{W_{ps}^{inter} W_{sq}^{inter}}{(\sum_k W_{ks}^{inter})(\sum_k W_{kq}^{inter})} \end{aligned} \quad (3)$$

And $p^{(2)}(t_p|t_q)$ can be computed analogously, which is equal to $p^{(2)}(i_p|i_q)$ actually. This result shows that intra-similarity is dependent on inter-similarity. So we complete the proof.

Theorem 1 tells us that if the projection is inter-similarity preserved, then it is also intra-similarity preserved according to formula (3). In other words, actually we just need to take inter-similarity preserving into consideration for CVHFL.

### 3.3. Kernelized Hashing for Cross-View

Kernel function $\kappa : \mathbf{R}^d \times \mathbf{R}^d \mapsto \mathbf{R}$ is often utilized to tackle linearly inseparable problems for classification. Actually, inter-similarity preserving in CVHFL is almost equivalent to classification problem. Suppose that the $t$-th bit of codewords for $\mathbf{x}_i$ is $-1$, i.e. $h_{\mathcal{X}}^{(t)}(\mathbf{x}_i) = -1$. According to Definition 1, a inter-similarity preserved hashing function should map $h_{\mathcal{Y}}^{(t)}(\mathbf{y}_i)$ to $-1$. In other words, $\{\mathbf{y}_i\}$ should be labeled by $h_{\mathcal{X}}^{(t)}(\mathbf{x}_i)$ when learning classification hyperplanes. Moreover, because multiple distributions are involved, kernelized model performs better than linear model on cross-view datasets.

**Definition 2.** *Let* $\mathcal{H}$ *be the Reproducing Kernel Hilbert Space (RKHS) endowed with the kernel* $\kappa(\cdot,\cdot)$*, we define an integral operator* $L_f : \mathcal{H} \to \mathcal{H}$ *on data set* $\mathcal{Z}$ *as*

$$L_f(\cdot) = \frac{1}{|\mathcal{Z}|} \sum_{\mathbf{z} \in \mathcal{Z}} \kappa(\mathbf{z}, \cdot) f(\mathbf{z}) \quad (4)$$

According to Theorem 1, we only consider the inter-similarity between $\mathbf{x}_i$ and $\mathbf{y}_j$ based on $L_f$ as

$$\begin{aligned} g(\mathbf{x}_i, \mathbf{y}_j) &= \frac{1}{n^2} \sum_{k=1}^{n} \kappa(\mathbf{x}_k, \mathbf{x}_i)\kappa(\mathbf{y}_k, \mathbf{y}_j) f(\mathbf{x}_k) f(\mathbf{y}_k) \\ &= < L_f(\mathbf{x}_i), L_f(\mathbf{y}_j) >_{\mathcal{H}} \end{aligned} \quad (5)$$

However, computing (5) is quite time consuming because all data points in dataset are involved. Fortunately [16] indicates that $L_f$ can be approximate by $\widehat{L}_f$ defined as follows,

$$\widehat{L}_f(\cdot) = \frac{1}{|\widehat{\mathcal{Z}}|} \sum_{\widehat{\mathbf{z}} \in \widehat{\mathcal{Z}}} \kappa(\widehat{\mathbf{z}}, \cdot) f(\widehat{\mathbf{z}}) \quad (6)$$

where $\widehat{\mathcal{Z}} \subset \mathcal{Z}$ is randomly sampled from $\mathcal{Z}$, usually $|\widehat{\mathcal{Z}}| \ll |\mathcal{Z}|$. In other words, with high probability, for any $\mathbf{x}_i$ and $\mathbf{y}_j$, $| < L_f(\mathbf{x}_i), L_f(\mathbf{y}_j) >_{\mathcal{H}} - < \widehat{L}_f(\mathbf{x}_i), \widehat{L}_f(\mathbf{y}_j) >_{\mathcal{H}} |$ should be small when $|\widehat{\mathcal{Z}}|$ is sufficiently large.

### 3.4. Overall Objective Function

Based on Theorem 1, we just consider inter-similarity for CVHFL. The objective function of KSH-CV share the same form as (1), but the difference is that KSH-CV maximizes the weighted inter-similarity in kernelized space as follows:

$$\underset{\{f(\widehat{\mathbf{x}})\}_{\widehat{\mathbf{x}} \in \widehat{\mathcal{X}}}, \{f(\widehat{\mathbf{y}})\}_{\widehat{\mathbf{y}} \in \widehat{\mathcal{Y}}}}{\text{maximize}} \sum_{i,j=1}^{n} W_{ij} \widehat{g}(\mathbf{x}_i, \mathbf{y}_j) \quad (7)$$

where $W$ is the average inter-similarity matrix among objects and $\widehat{g}(\mathbf{x}_i, \mathbf{y}_j) = < \widehat{L}_f(\mathbf{x}_i), \widehat{L}_f(\mathbf{y}_j) >_{\mathcal{H}}$. $\widehat{\mathcal{X}} \subset \mathcal{X}$ ($\widehat{\mathcal{Y}} \subset \mathcal{Y}$) are randomly sampled from $\mathcal{X}(\mathcal{Y})$. $W$ can be constructed in unsupervised way just as many other SpH and supervised way:

$$W_{ij} = \delta_{ij}(\exp\{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{\sigma_i^{\mathcal{X}} \sigma_j^{\mathcal{X}}}\} + \exp\{-\frac{||\mathbf{y}_i - \mathbf{y}_j||^2}{\sigma_i^{\mathcal{Y}} \sigma_j^{\mathcal{Y}}}\}) \quad (8)$$

similar to local scaling [17], we denote $\sigma_i^{\mathcal{X}}$ ( $\sigma_i^{\mathcal{Y}}$) as the median distance between the $i$-th item and all datapoints in $\mathcal{X}$ ($\mathcal{Y}$). Let $\delta_{ij} = 1$ if object $o_i$ and $o_j$ share the same labels and -1 otherwise, $\forall i, j$. Therefore, we add the semantic supervised information to affinity matrix. Furthermore, if set $\delta_{ij} = 1, \forall i, j$, then KSH-CV falls in an unsupervised method.

Actually, optimizing (7) would lead to trivial solution, i.e. $f(\widehat{\mathbf{x}})$ or $f(\widehat{\mathbf{y}})$ goes to infinity. Moreover, optimizing (7) is not intuitive. Instead we rewrite it to matrix form as follows,

$$\begin{aligned} \underset{\mathbf{a}_{\mathcal{X}}, \mathbf{a}_{\mathcal{Y}}}{\text{maximize}} \quad & \mathbf{a}_{\mathcal{X}}^T K(\widehat{\mathcal{X}}, \mathcal{X}) W K^T(\widehat{\mathcal{Y}}, \mathcal{Y}) \mathbf{a}_{\mathcal{Y}} \\ s.t. \quad & \mathbf{a}_{\mathcal{X}}^T \mathbf{a}_{\mathcal{X}} = 1, \mathbf{a}_{\mathcal{Y}}^T \mathbf{a}_{\mathcal{Y}} = 1 \end{aligned} \quad (9)$$

where $m = |\widehat{\mathcal{X}}|(|\widehat{\mathcal{Y}}|)$, $\mathbf{a}_{\mathcal{X}} = [f(\widehat{\mathbf{x}}_1), ..., f(\widehat{\mathbf{x}}_m)], \mathbf{a}_{\mathcal{Y}} = [f(\widehat{\mathbf{y}}_1), ..., f(\widehat{\mathbf{y}}_m)]$. The kernel matrix $K(\widehat{\mathcal{X}}, \mathcal{X}) \in \mathbf{R}^{m \times n}$ is defined as $K(\widehat{\mathcal{X}}, \mathcal{X})_{ij} = \kappa(\widehat{\mathbf{x}}_i, \mathbf{x}_j)$, and $K(\widehat{\mathcal{Y}}, \mathcal{Y})$ is defined analogously. The normalization constraints are added to avoid trivial solution. Then the hash function can be obtained:

$$h_{(\cdot)}(\mathbf{z}) = \text{sgn}(\mathbf{a}^{*T}_{(\cdot)} \mathbf{z}). \quad (10)$$

where $\text{sgn}(u) = 1$ if $u > 0$ and $-1$ otherwise for all $u \in \mathbf{R}$, and $\mathbf{a}^{*T}_{(\cdot)}$ is the optimal solutions of formula (9).

### 3.5. Solution for KSH-CV

Solving the formula (9) can only learn one hash function for one bit. To learn compact hash codes for multiple bits is still a challenging problem. A solution is to use singular value decomposition (SVD) on matrix $K(\widehat{\mathcal{X}}, \mathcal{X}) W K^T(\widehat{\mathcal{Y}}, \mathcal{Y})$, and then select the largest $k$ singular value with their corresponding singular vectors as mapping vectors $\{\mathbf{a}_{\mathcal{X}}^{(i)}\}_{i=1}^{k}$ and

$\{\mathbf{a}_{\mathcal{Y}}^{(i)}\}_{i=1}^{k}$ respectively. However, this solution has the same shortcomings as SpH model, that is, suffers instability among datasets as very low variance directions maybe picked [7].

Adaboost is another method to learn multi-bit compact codewords for CVHFL [14, 13]. It minimizes the exponential error of classification $\sum_i \exp(-s_i F_k(\mathbf{z}_i))$, where $s_i \in \{-1, 1\}$ are the target values of the training data, and $F_k(\mathbf{z}) = \text{sgn}(\sum_{m=1}^{k} \alpha_m f_m(\mathbf{z}))$ is the final classifiers, in which $f_m(\mathbf{z})$ is the basis classifier. But the optimization objective for CVHFL differs from classification. In this paper, we define the exponential mapping loss for CVHFL, which measures the mapping quality for inter-similarity preserving.

**Definition 3.** *The exponential mapping loss for cross-view similarity search is defined as*

$$E = \sum_{ij} \exp\{\delta_{ij} D_h(\mathbf{x}_i, \mathbf{y}_j)\} \tag{11}$$

*where $D_h(\cdot, \cdot)$ denotes the Hamming distances.*

Literature [3] shows that:

$$D_h(\mathbf{x}_i, \mathbf{y}_j) \propto (F_{\mathcal{X}}^{(m)}(\mathbf{x}_i))^T (F_{\mathcal{Y}}^{(m)}(\mathbf{y}_j)) \tag{12}$$

where $F_{\mathcal{X}}^{(m)}(\mathbf{x}) = [h_{\mathcal{X}}^{(1)}(\mathbf{x}), ..., h_{\mathcal{X}}^{(m)}(\mathbf{x})]^T$ is a codeword of $\mathbf{x}$, and the definition of $F_{\mathcal{Y}}^{(m)}(\mathbf{y})$ is analogous.

**Theorem 2.** *Relaxating the sign function of (10), the Alg. 1 minimises the exponential mapping loss for CVHFL.*

The Theorem 2 can be derived analogously to standard Adaboost (e.g. Section 14.3 in [18]).

**Algorithm 1** AdaBoost KSH-CV

**Input:**

Two kernel matrix $K(\widehat{\mathcal{X}}, \mathcal{X}), K(\widehat{\mathcal{Y}}, \mathcal{Y})$

Similarity matrix $W$, and the constant coefficients $\alpha_0$

**Output:**

Projection vector $\mathbf{a}_{\mathcal{X}}^{(i)}, \mathbf{a}_{\mathcal{Y}}^{(i)}, i = 1, ..., k$

1: Initialize the data weighting coefficients matrix $\Omega^{(1)}$ by setting each matrix elements as $1/n^2$

2: **for** $r = 1, ..., k$ **do**

3: Maximize formula:

$$\mathbf{a}_{\mathcal{X}}^T \{(K(\widehat{\mathcal{X}}, \mathcal{X}) W K^T(\widehat{\mathcal{Y}}, \mathcal{Y})) \odot \Omega^{(r)}\} \mathbf{a}_{\mathcal{Y}} \tag{13}$$

by SVD and get $\mathbf{a}_{\mathcal{X}}^{(r)}, \mathbf{a}_{\mathcal{Y}}^{(r)}$, where the symbol $\odot$ represents the Hadamard product.

4: Update weighting matrixes according to

$$\Omega_{ij}^{(r+1)} = \Omega_{ij}^{(r)} \exp\{-\alpha_0 I(p_{ij}^{(r)})\} \tag{14}$$

where $I(\cdot)$ is the indicator function, and $p_{ij}^{(r)}$ denotes $\delta_{ij} = (h_{\mathcal{X}}^{(r)}(\mathbf{x}_i) h_{\mathcal{Y}}^{(r)}(\mathbf{y}_j))$, and is normalized by sum.

5: **end for**

**Table 1**. mAP of KSH-CV on Three Datasets

| # bits | Wiki | | NUS-WIDE | | MIRFLICKR-25000 | |
|---|---|---|---|---|---|---|
| | Spv & SVD | Spv & Bst | Spv & SVD | Spv & Bst | Spv & SVD | Spv & Bst |
| | Task:Image → Text | | Task:Image → Text | | Task:SIFT → CEDD | |
| 12 | 29.08% | **31.15%** | 49.54% | **51.78%** | 62.69% | **65.38%** |
| 24 | 27.54% | **32.70%** | 48.78% | **53.66%** | 59.63% | **65.71%** |
| 32 | 28.39% | **30.74%** | 47.68% | **54.01%** | 60.08% | **64.73%** |
| 48 | 30.68% | **31.36%** | 47.45% | **50.59%** | 51.46% | **62.45%** |
| | Task:Text → Image | | Task:Text → Image | | Task:CEDD → SIFT | |
| 12 | 23.48% | **23.84%** | 55.48% | **57.87%** | 58.04% | **62.16%** |
| 24 | 21.08% | **22.95%** | **58.27%** | 57.17% | 58.66% | **64.50%** |
| 32 | 20.67% | **23.64%** | **57.00%** | 55.42% | 57.56% | **63.53%** |
| 48 | 20.71% | **22.91%** | 55.10% | **56.89%** | 59.98% | **63.31%** |

## 4. EXPERIMENTS

We evaluated KSH-CV on three different datasets for cross-view similarity search: Wiki (2,866 objects), NUS-WIDE (186,577 objects), and MIRFLICKR-25000 (25,000 objects).

### 4.1. Datasets

**Wiki**[1]. This dataset was collected from Wikipedia. It consists of 2,866 image-text pairs. Each image is represented by 128-dimension SIFT histograms and each text is represented by 10-dimension topics vector. It contains 10 semantic classes and each pair is labeled with one of them. We use 75% of the pairs as the training set, the remaining 25% as the query set.

**NUS-WIDE**[2].This dataset is a real-world web image database containing 81 concepts and 269,648 images with tags. In our experiment, we select ten largest concepts and the corresponding 186,577 images. Images are represented by 500-dimension SIFT histograms, and texts are represented by index vectors of the most frequent 1000 tags. Each pair is annotated by at least one of 10 concepts. Pairs are considered to be similar if they share at least one concepts. We use 99% of the data as the training set and the rest 1% as the query set.

**MIRFLICKR-25000**[3]. This dataset consists of 25,000 images, and every image is annotated by some potential or relevant labels from 38 unique labels. Images are described by 100-dimension SIFT histograms which mainly encode surface texture, and 144-dimension CEDD features [8] which focus on color and edge directivity. Because they show similarity in texture space while they are dissimilar in color space, we choose them to simulate a cross-view setting as in [19].

### 4.2. Experimental Setup

In our experiments, we compared KSH-CV to four state-of-the-art hashing methods for cross-view, i.e., CMSSH [13] [4], CHMIS [9] [4] , IMH [12] [1] [5]and CVH [10] [5] . We have two versions of KSH-CV named as KSH-CV$_{\text{Spv \& Bst}}$ and

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

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

[3] http://press.liacs.nl/mirflickr/

[4] The source code is kindly provided by the authors.

[5] We implemented it ourselves because the code is not publicly available.

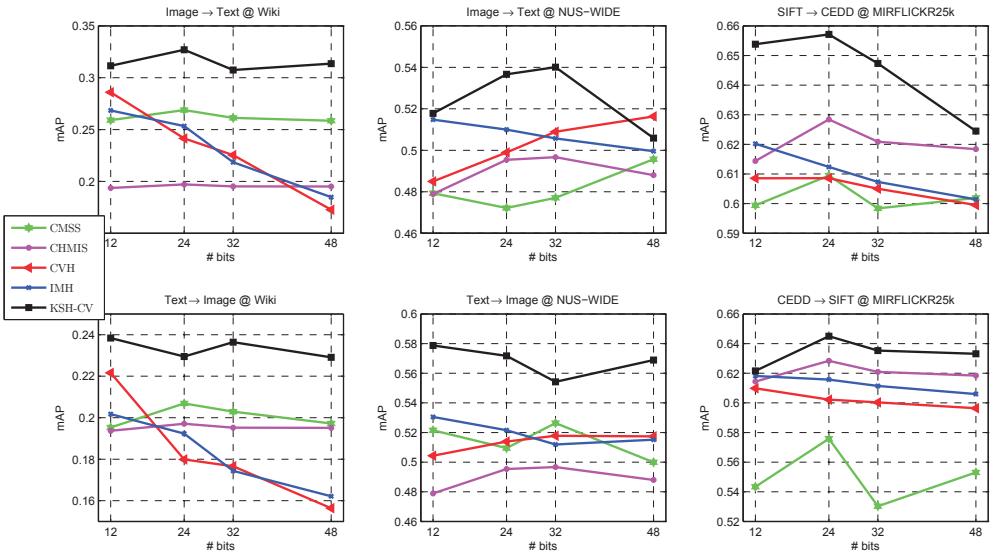


**Fig. 1**. Compare mAP on Wiki, NUS-WIDE and MIRFLICKR-25k

KSH-CV$_{\text{Spv \& SVD}}$ respectively. KSH-CV$_{\text{Spv \& Bst}}$ learns compact hashcodes by Adaboost while KSH-CV$_{\text{Spv \& SVD}}$ learns hash projector by selecting the largest $k$ singular values and their corresponding singular vectors. Both versions construct similarity matrix in a supervised way. And we regard KSH-CV as KSH-CV$_{\text{Spv \& Bst}}$, when no ambiguity is intrigued.

The performance measure is the mean average precision (mAP) [14], which is the mean of average precision (AP), and AP of top $R$ retrieved documents is defined as: $\text{AP} = \frac{1}{L}\sum_{i=1}^{R} P(i) \times \delta(i)$, where $P(i)$ denotes the precision of the top $i$ retrieved documents, and $\delta(i)$ is an indicator function which is equals to 1 if the item at rank $i$ is a relevant instance or 0 otherwise, and $L$ is the number of relevant documents in the retrieved set.

We chose the Gaussian RBF kernel $\kappa(\mathbf{x}, \mathbf{y}) = \exp(-||\mathbf{x} - \mathbf{y}||/2\sigma^2)$ as kernel function, and set $m = 300$, $R = 50$, $\alpha_0 = 0.2$ on each dataset. We select 10000 datapoints from training set randomly (except for Wiki) to train KSH-CV. Considering that CVH, CMSS,IMH and CHMIS are not scalable for large-scale datasets, we used the same subset as utilized in KSH-CV as training set on NUS-WIDE and MIRFLICKR25k. All results are averaged over 5 runs.

### 4.3. Experimental Results

To verify the influence of Adaboost algorithm on KSH-CV, we compared KSH-CV$_{\text{Spv \& Bst}}$ to KSH-CV$_{\text{Spv \& SVD}}$ on each database firstly. The experiment results are shown in Table 1, and we bolded better results of two versions of KSH-CV.

**SVD vs. Adaboost**. Generally, KSH-CV$_{\text{Spv \& Bst}}$ achieves better performance over all databases with NUS-WIDE becoming an exception when codewords of text as query. As mentioned above, Adaboost aims to minimize exponential mapping loss for cross-view (11) at each iteration, and avoids picking low variance directions compare with SVD.

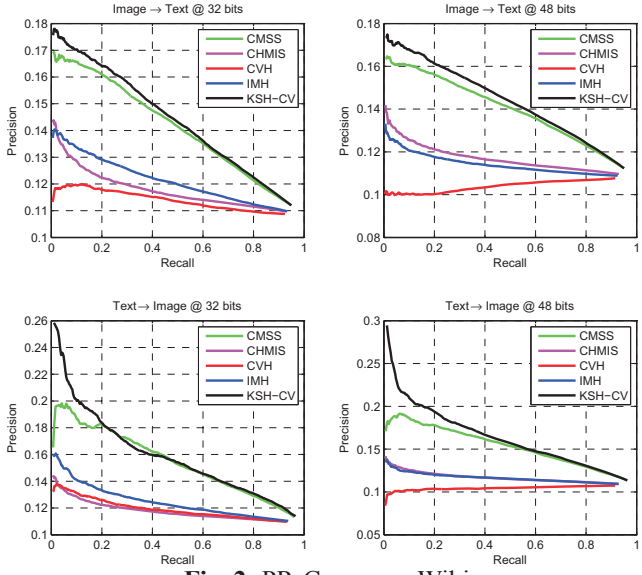


**Fig. 2**. PR-Curves on Wiki

**KSH-CV vs. Baslines.** We further compared KSH-CV with other four state-of-the-art cross-view models in Fig. 1. We can see that KSH-CV significantly improves the performance over baseline methods. Actually, CHMIS, IMH and CVH can be regarded as eigenvalue-decomposition-based and unsupervised models, while CMSS can be regarded as boosted and supervised method. Moreover, IMH and CVH preserving intra- and inter-similarity simultaneously. At least three conclusions can be drawn from the experiment results. Firstly, CMSS achieves better performance than CHMIS and CVH on Wiki, but the situation is opponent on MIRFLICKR-25k. This means that supervised information is more precious when source data lacking inter-similarity. Secondly, CMSS, which lacks kernelized technique, shows worse performance on every dataset than KSH-CV. Finally, eigenvalue-decomposition-based model shows instable performance on

**Table 2**. Effects of training size on MAP performance

| Size | NUS-WIDE | | MIRFLICKR-25k | |
|---|---|---|---|---|
| | Image→Text | Text→Image | SIFT→CEDD | CEDD → SIFT |
| 1k | 49.49% | 49.94% | 61.31% | 60.67% |
| 5k | 50.46% | 52.10% | 62.27% | 60.69% |
| 10k | **50.59%** | **56.89%** | **62.45%** | **63.31%** |

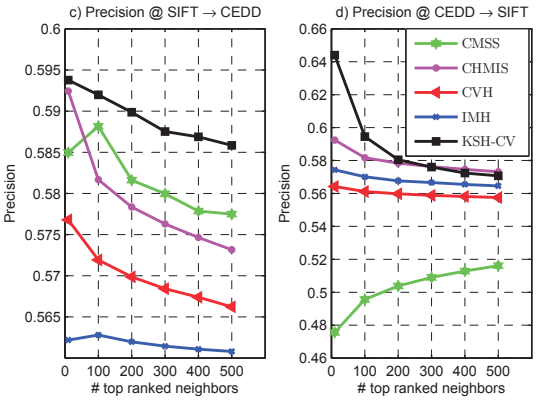


**Fig. 3**. Precision on MIRFLICKR-25K @ 48bits.

Wiki, which verifies our viewpoint that directions with low variance may be picked with the increase of bit number.

The precision-recall curves on Wiki are shown in Fig.2, and Fig.3 shows precision vs. num of top ranked neighbors. Like mAP, KSH-CV outperforms other hashing methods.

**Data Size.** Last but not the least, we vary the size of training data from 1k to 10k to verify the stabilization of KSH-CV, and the result is shown in Table 2. Obviously, the larger the training data size is, the more effective hash functions can be learnt. And we can observe that, when the data size increases from 5k to 10k, the effectiveness gains are less than 1k to 5k.

## 5. CONCLUSIONS

In this paper, kernel-based supervised hashing for cross-view (KSH-CV), utilizing kernelized and supervised information, is proposed as a novel SpH method for cross-view hash-based function learning. Based on exponential mapping loss, KSH-CV is boosted differently from the standard Adaboost algorithm and other boosted-based hashing models. Experiments show that supervised-based models performs better if different views share more similarity. And the performance of spectral-based models shows instability when the bit number increases. It is also shown that KSH-CV outperforms state-of-the-art cross-view hashing methods on three cross-view datasets, which validates the effectiveness of our method.

## 6. ACKNOWLEDGMENTS

This research was supported by the National Basic Research Project of China (Grant No. 2011CB70700), the National Natural Science Foundation of China (Grant No. 61271394), and the National HeGaoJi Key Project (No. 2013ZX01039-002-002). And the authorswould like to thank the reviewers for their valuable comments.

## 7. REFERENCES

[1] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al., "Similarity search in high dimensions via hashing," in *VLDB*, 1999. 1, 2, 4

[2] Yair Weiss, Antonio Torralba, and Rob Fergus, "Spectral hashing," NIPS, 2008. 1, 2

[3] Wei Liu, Jun Wang, Rongrong Ji, Yu-Gang Jiang, and Shih-Fu Chang, "Supervised hashing with kernels," in *CVPR*. IEEE, 2012. 1, 2, 4

[4] Brian Kulis and Kristen Grauman, "Kernelized locality-sensitive hashing for scalable image search," in *ICCV*. IEEE, 2009. 1, 2

[5] Yunchao Gong and Svetlana Lazebnik, "Iterative quantization: A procrustean approach to learning binary codes," in *CVPR*. IEEE, 2011. 1

[6] Kaiming He, Fang Wen, and Jian Sun, "K-means hashing: an affinity-preserving quantization method for learning binary compact codes," in *CVPR*, 2013. 1

[7] Jun Wang, Sanjiv Kumar, and Shih-Fu Chang, "Semi-supervised hashing for scalable image retrieval," in *CVPR*. IEEE, 2010. 1, 4

[8] Savvas A Chatzichristofis and Yiannis S Boutalis, "Cedd: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval," in *Computer Vision Systems*. Springer, 2008. 1, 4

[9] Dan Zhang, Fei Wang, and Luo Si, "Composite hashing with multiple information sources," in *SIGIR*. ACM, 2011. 1, 2, 4

[10] Shaishav Kumar and Raghavendra Udupa, "Learning hash functions for cross-view similarity search," in *IJCAI*. AAAI Press, 2011. 1, 2, 4

[11] Saehoon Kim, Yoonseop Kang, and Seungjin Choi, "Sequential spectral learning to hash with multiple representations," in *ECCV*. Springer, 2012. 1, 2

[12] Jingkuan Song, Yang Yang, Yi Yang, Zi Huang, and Heng Tao Shen, "Inter-media hashing for large-scale retrieval from heterogeneous data sources," in *ICMD*. ACM, 2013. 1, 2, 4

[13] Michael M Bronstein, Alexander M Bronstein, Fabrice Michel, and Nikos Paragios, "Data fusion through cross-modality metric learning using similarity-sensitive hashing," in *CVPR*. IEEE, 2010. 2, 4

[14] Yi Zhen and Dit-Yan Yeung, "Co-regularized hashing for multimodal data," in *NIPS*, 2012. 2, 4, 5

[15] Fan RK Chung, *Spectral graph theory*, AMS Bookstore, 1997. 2

[16] Steve Smale and Ding-Xuan Zhou, "Geometry on probability spaces," *Constructive Approximation*, 2009. 3

[17] P Perona and L Zelnik-Manor, "Self-tuning spectral clustering," *NIPS*, 2004. 3

[18] Christopher M Bishop et al., *Pattern recognition and machine learning*, Springer, New York. 4

[19] Novi Quadrianto and Christoph H Lampert, "Learning multi-view neighborhood preserving projections," in *ICML*, 2011. 4