TUCH: Turning Cross-view Hashing into Single-view Hashing via Generative Adversarial Nets *

Xin Zhao†, Guiguang Ding†, Yuchen Guo†, Jungong Han‡, Yue Gao†
†Tsinghua National Laboratory for Information Science and Technology (TNList)
School of Software, Tsinghua University, Beijing 100084, China
‡School of Computing & Communications, Lancaster University, UK
{zhaoxin19,yuchen.w.guo}@gmail.com, {dinggg,gaoyue}@tsinghua.edu.cn, jungong.han@northumbria.ac.uk

Abstract

Cross-view retrieval, which focuses on searching images as response to text queries or vice versa, has received increasing attention recently. Cross-view hashing is to efficiently solve the cross-view retrieval problem with binary hash codes. Most existing works on cross-view hashing exploit multi-view embedding method to tackle this problem, which inevitably causes the information loss in both image and text domains. Inspired by the Generative Adversarial Nets (GANs), this paper presents a new model that is able to Turn Cross-view Hashing into single-view hashing (TUCH), thus enabling the information of image to be preserved as much as possible. TUCH is a novel deep architecture that integrates a language model network T for text feature extraction, a generator network G to generate fake images from text feature and a hashing network H for learning hashing functions to generate compact binary codes. Our architecture effectively unifies joint generative adversarial learning and cross-view hashing. Extensive empirical evidence shows that our TUCH approach achieves state-of-the-art results, especially on text to image retrieval, based on image-sentences datasets, i.e. standard IAPRTC-12 and large-scale Microsoft COCO.

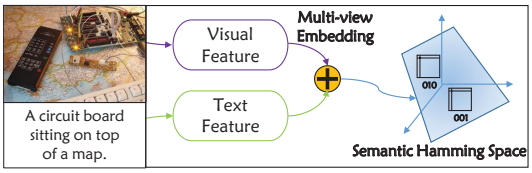
1 Introduction

While multimedia big data of massive volumes and high dimensions are pervasive in search engines and social networks, it has attracted increasing attention to approximate nearest neighbors search across different media modalities that brings both computation efficiency and search quality. Since correspondence data from different modalities may endow semantic correlations, it has a tendency to support cross-view retrieval that returns relevant search results from one view as response to query of another view, e.g. retrieval of images of text query. An effective solution to cross-view retrieval is hashing method that learns compact binary codes with similar binary codes for similar objects from high-dimensional data. This paper focuses on cross-view hashing that builds isomorphic hash codes for efficient cross-view retrieval. So far, effective and efficient cross-view hashing remains a big challenge, because of the heterogeneity across different views [Wei *et al.*, 2014], and the semantic gap between low-level features and high-level semantics [Smeulders *et al.*, 2000].

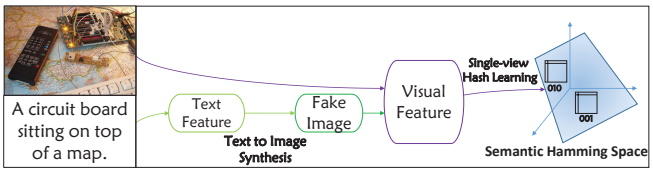
Several recent models for cross-view hashing [Bronstein *et al.*, 2010; Zhen and Yeung, 2012; Lin *et al.*, 2015; 2016; Wu *et al.*, 2015; Cao *et al.*, 2016a; Ding *et al.*, 2016; Guo *et al.*, 2016] have followed the same multi-view embedding framework, which integrates both image data and text data into an independent semantic embedding space. The multi-view embedding framework basically learns projection matrixes from image space and text space to the embedding space as Figure 1 (a). Due to this unnecessary intermediate transformation, the information leak is unavoidable. Intuitively, a single-view problem may be easier to solve than a cross-view problem. And if we can convert a cross-view hashing problem into a single-view hashing problem on a specific domain, at least the information loss on this domain will be minimized. Generative Adversarial Networks (GANs) approach is originally used for text to image synthesis [Reed *et al.*, 2016], which provides a ready-to-use solution to convert text data into image domain, making it possible to solve cross-view hashing problem with single-view method.

In this work, we strive for the goal of efficient cross-modal retrieval of images in response to natural sentence queries or vice versa. Inspired by the idea of GANs, this paper presents a new model which turns cross-view hashing into single-view hashing on image domain with NO multi-view embedding required, whose overview is illustrated in Figure 1 (b). It will keep the information of image as much as possible and result in good aggregation characteristics of image features so that we can achieve a better retrieval result in cross-view retrieval. TUCH architecture constitutes a language model network T for text feature extraction, a generator network G to generate fake images from text feature and a discriminate hashing network H for learning hashing functions, eventually generating compact binary codes, while distinguishing correlation of image and text. Net G and net H constitute an end-to-end multi-task deep learning model to train generative adversarial model and hashing function simultaneous-

*This research was supported by the National Natural Science Foundation of China (No. 61571269) and the Royal Society Newton Mobility Grant (IE150997). Corresponding author: Guiguang Ding.



(a) Multi-view Embedding Framework



(b) TUCH Framework

Figure 1: Multi-view hashing framework embeds both image data and text data into an independent semantic hamming space while TUCH framework generates fake image with text to image synthesis method and applies single-view hashing method on the image domain (with generated fake images).

ly. Most of the deep hashing models for image retrieval tried to add a regularization term to minimize the quantization error into loss function [Lai *et al.*, 2015; Zhu *et al.*, 2016; Cao *et al.*, 2016b] or insert a gradient snapping layer into the network to achieve it [Liu and Lu, 2016]. But the quantization error cannot be avoided in those frameworks. We add a discrete layer in our TUCH framework which directly outputs binary codes for loss computation and adopt a simple backpropagation algorithm for optimization. In summary, we make the following contributions in this paper:

- We put forward a novel end-to-end deep learning approach termed as TUCH for cross-view hashing, which is the first to transfer multi-view problem into a single-view hashing problem in image domain with GANs.
- A multi-task architecture is adopted considering the tag information from the retrieval similarity and the description information for sentence to image generation.
- We add a discrete layer in TUCH, which directly outputs binary codes via loss computation. By doing so, the information loss during the feature quantization is avoided, which is unachievable in previous works. A simple and efficient algorithm is adopted for loss backpropagation at the discrete layer.

2 Related Work

2.1 Hashing for Cross-View Retrieval

Hashing is a widely used indexing technique for image retrieval [Wang *et al.*, 2016]. Locality Sensitive Hashing [Gionis *et al.*, 1999] can be regarded as the seminal hashing work which adopts random splits in the feature space to generate binary codes. Thereafter, a number of learning based hashing approaches are proposed, which fall into two categories: unsupervised hashing and supervised hashing. No supervision but only statistics information of data is taken into consideration in unsupervised hashing approaches, such as manifold structure [Weiss *et al.*, 2008; Liu *et al.*, 2014; Shen *et al.*, 2015b], variance of features [Gong *et al.*, 2013; Xu *et al.*, 2013; Kong and Li, 2012] and cluster property [He *et al.*, 2013]. On the contrary, supervised hashing approaches take advantage of the supervised knowledge so as to better capture the intrinsic semantic property of data, in which the representatives include Supervised Hashing with Kernels [Liu *et al.*, 2012], Supervised Discrete Hashing [Shen *et al.*, 2015a] and Latent Factor Hashing [Zhang *et al.*, 2014].

Several recent models for cross-view hashing [Bronstein *et al.*, 2010; Zhen and Yeung, 2012; Lin *et al.*, 2015; Wu *et al.*, 2015] have followed the multi-view embedding framework, which embeds both image data and text data into an independent semantic embedding space. The multi-view embedding framework basically learns projection matrixes from image space and text space to the embedding space.

Recent years, researchers are trying to apply deep neural networks in hashing problem. A two-stage training strategy was proposed that generates hash codes via a combination of disjoint CNN network and discrete code learning [Xia *et al.*, 2014]. Alternatively, some end-to-end deep hashing models for image retrieval were proposed by minimizing the quantization error [Lai *et al.*, 2015; Zhu *et al.*, 2016] or inserting a gradient snapping layer into the network [Liu and Lu, 2016] while simultaneously minimizing the sample similarity loss. Benefiting from the power of CNN, these approaches achieve significant improvement over the traditional approaches, but they can only be applied to single-view retrieval. DVSH model [Cao *et al.*, 2016a] is the first end-to-end deep learning approach for cross-view hashing that enables efficient cross-view retrieval of images in response to sentence queries and vice versa. Following the multi-view embedding framework, it applies deep hashing method in cross-view retrieval problem and achieves the state-of-the-art results. However, the projection from image view and text view to the semantic embedding space will cause the information loss in both views and poor cross-view retrieval result. That is to say, both deep and non-deep multi-view embedding based cross-view hashing methods still have room for improvement.

2.2 Generative Adversarial Networks (GANs)

Two challenges in multi-modal learning include: 1) learning a shared representation across modalities, and 2) estimating missing data (e.g. by retrieval or synthesis) in one modality conditioned on another. Generative adversarial networks (GANs) [Goodfellow *et al.*, 2014] have benefited from convolutional decoder networks for the generator network module. A Laplacian pyramid of adversarial generator and discriminators is used to synthesize images at multiple resolutions [Denton *et al.*, 2015]. A standard convolutional decoder incorporating batch normalization is adopted [Radford *et al.*, 2015], which developed a highly effective and stable architecture to achieve striking image synthesis results. An end-to-end differentiable architecture from the character level to pixel level is proposed using model conditions on text descriptions to achieve sentence text to corresponding image synthesis [Reed *et al.*, 2016], which provide a good way to convert sentence text into fake image, making it possible to

transfer a cross-view sentence-image retrieval problem into a single-view problem. In this work we adopt a multi-task GAN architecture which aims to generate fake image with tag features from sentence text, so that we can embed text domain into image domain for a single-view hashing method.

3 Background

3.1 Generative Adversarial Networks (GANs)

Generative adversarial networks (GANs) consist of a generator G and a discriminator D that compete in a two-player minimax game: The discriminator tries to distinguish real training data from synthetic images, while the generator tries to fool the discriminator. Specifically, D and G play the following game on V(D,G):

$$\begin{aligned}\max_G \min_D V(D,G) = E_{x\sim p_{data}(x)}[log(D(x))] \\ +E_{z\sim p_g(z)}[log(1-D(G(z)))]\end{aligned} \tag{1}$$

where x represents image sample from dataset, z represents random noise vetor. It is proved that this minimax game has a global optimum precisely when $p_g(z) = p_{data}$, and that under mild conditions (e.g. G and D have enough capacity) $p_g(z)$ converges to p_{data}.

3.2 Conditional Generative Adversarial Networks for Text to Image Synthesis

Conditional Generative Adversarial Networks have been used for text to image synthesis by small modification over GANs [Reed *et al.*, 2016] as below:

$$\begin{aligned}\max_G \min_D V(D,G) = E_{x\sim p_{data}(x)}[log(D(x,t))] \\ +\lambda_1 E_{x\sim p_{data}(x)}[1-log(D(\hat{x},t))] \\ +\lambda_2 E_{z\sim p_g(z)}[log(1-D(G(z,t),t))]\end{aligned} \tag{2}$$

where t represents the feature of sentence text, x refers to true image sample relevant to t, $\hat{x}$ represents image sample inrelevant. In brief, generator generates a fake image with random noise and text feature to fool the discriminator, and the discriminator tries to distinguish real text-relevant training data from synthetic images and irrelevant ones.

4 Turning Cross-view Hashing into Single-view Hashing via GANs

4.1 Problem Definition

The definition of cross-view hashing between image and sentence based text in this paper follows [Cao *et al.*, 2016a]. We are given n training images$\{I_1,\ldots,I_n\}$ and each image I_m has k_m^t visual concept tags and a set of k_m^d sentence descriptions $\mathcal{D}_m = \{D_1,\ldots,D_{k_m^d}\}$. Each visual concept tag belongs to set $\mathcal{T} = \{T_1,\ldots,T_K\}$, where K is the size of $\mathcal{T}$. For each image there is a label vector $y_m \in \{0,1\}^K$ where $y_{mj} = 1$ if I_m has tag T_j and $y_{mj} = 0$ otherwise. The set of all descriptions is $\mathcal{D} = \bigcup_{i=1}^n D_i$. We aim to learn hashing models $h^I : I \to \{-1,1\}^{K_H}$ and $h^D : \mathcal{D} \to \{-1,1\}^{K_H}$ to map image and sentence to K_H-bit codes. It is desired that h^I and h^D can preserve the semantic similarity in visual and text domains in the same Hamming space $\{-1,1\}^{K_H}$.

4.2 Network Architecture

We use the following notation. The generator network is denoted G: $R^Z \times R^{K_t} \to R^I$. The discriminative hashing network as H: $R^I \times R^{K_t} \to \{0,1\} \times R^K$, where K_t is the dimension of the text description encoding. I is the dimension of the image, and Z is the dimension of the noise input to G. We illustrate our network architecture in Figure 2.

In generator G, first we sample from the noise prior $z \in R^Z \sim \mathcal{N}(0,1)$ and we concatenate the text query encoding t with z. Following this, inference proceeds as in a normal deconvolutional network: we feed-forward it through the generator G; a synthetic image $\dot{x}$ is generated via $\dot{x} \leftarrow G(z,t)$. Image generation corresponds to feed-forward inference in the generator G conditioned on query text and a noise sample.

In the discriminative hashing network H, we perform a CNN network with multi-task objective functions. There are two input branches,, where the first one is an image branch with a standard 18 layers Deep Residual Network (Resnet) [He *et al.*, 2016], while the other is a fully-connected text branch. The discrete layer which outputs binary hash codes follows right after the last fully-connected layer (FC-I) of the image branch. The semantic embedding loss is calculated by the discrete layer. We concatenate the last layer of text branch with FC-I and put a discriminative loss after it. The semantic embedding loss is to preserve the similarity of both image domain and text domain. The discriminative loss is to judge if an image (real or fake) is relevant to the given text encoding.

4.3 Loss Function

Semantic Embedding Loss. Given an input image I_m, suppose the output of the hashing layer $h_m \in \{-1,1\}^{K_H}$ is the K_H-bit hash codes. Then the hash codes are projected to the final semantic vector space by the parameter W_h and the output is denoted as ϕ_m. The semantic embedding loss mostly focuses on improving the generalize ability of the hashing model. Meanwhile, it is also desired that the samples with common concept have very similar hash codes whereas samples without common concept have dissimilar hash codes. The image branch of H can be regarded as a single-view hashing network. The objective function of G and H is the same. That is to say, G aims to generate similarity-preserving fake image and H aims to extract similarity-preserving hash codes. We take W_h as a projection matrix from Semantic Hamming space $\{-1,1\}^{K_H}$ to the tag space R^K. The semantic embedding loss is as follow:

$$\begin{aligned}\mathcal{L}^S(I) = -\sum_m \sum_{i=1}^K (y_{mi} log(\frac{e^{\phi_{mi}}}{1+e^{\phi_{mi}}}) \\ +(1-y_{mi}) log(\frac{1}{1+e^{\phi_{mi}}}))\end{aligned} \tag{3}$$

where y_m refers to the label vector of I_m, $\phi_m = W_h h_m \in R^K$, h_m is the K_H bit hash code of I_m.

Discriminative Loss. Given a text encoding t, suppose the image relevant to t is x, the image irrelevant to t is $\hat{x}$. The discriminative branch of H is referred as $H_D : R^I \times R^{K_t} \to \{0,1\}$.

The generator G tries to generate fake images to fool the discriminative branch of H. The discriminative branch of H

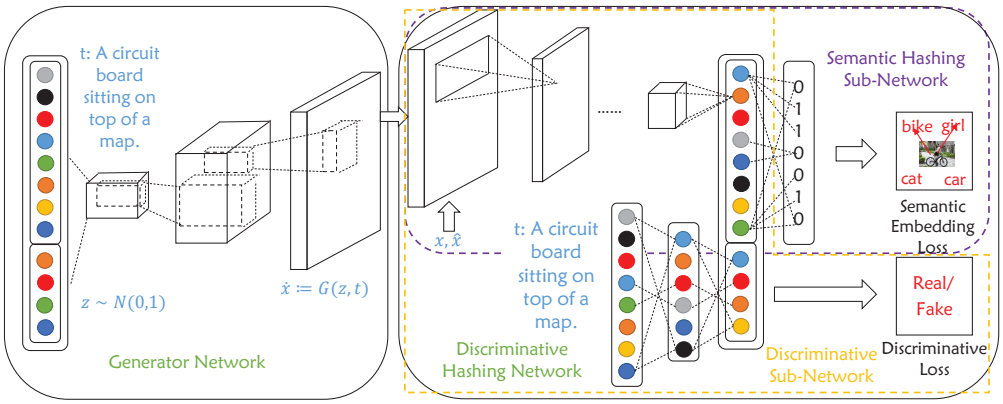


Figure 2: Our multi-task TUCH architecture. Generator network and discriminative sub-network constitute a normal conditional GANs. Semantic hashing sub-network is a normal image hashing network based on CNN. Text encoding t extracted by language model T is used in both generator network and discriminative hashing network. x is real relevant image, $\dot{x}$ is fake image generated by G with t,$\hat{x}$ is real irrelevant image.

tries to distinguish real relevant images from synthetic images and real irrelevant images. According to Equ (2), we can get $\mathcal{L}_H$ and $\mathcal{L}_G$ from $V(H_D, G)$, which donate the discriminative loss to be minimized for generator G and discriminative hashing network H.

$$\mathcal{L}_H^D = \lambda_2 log(1 - H_D(G(z,t),t)) + \lambda_1(1 - log(H_D(\hat{x},t))) + log(H_D(x,t)) \tag{4}$$

$$\mathcal{L}_G^D = log(H_D(G(z,t),t)) \tag{5}$$

The combined multi-task loss is as below:

$$\mathcal{L}_H = \mathcal{L}^S(\{x, \hat{x}, G(z,t)\}) + \lambda_H \mathcal{L}_H^D \tag{6}$$

$$\mathcal{L}_G = \mathcal{L}^S(\{G(z,t)\}) + \lambda_G \mathcal{L}_G^D \tag{7}$$

4.4 Optimization

With the overall loss function, we adopt the backpropagation algorithm with mini-batch stochastic gradient descent method to train the network. The optimization is a 2-step procedure, in which we optimize G and H in turn. Algorithm 1 summarizes the training procedure.

There is no other problem during backpropagation except that we adopt a discrete hashing layer in the network whose discretion operation by sign function is non-differentiable at 0 and the derivative at the other part is also zero such that the gradient vanishes when propagated through this layer. To address this issue, we adopt the straight-through estimator to compute the gradients. Specifically, suppose the input of the hashing layer is $r \in R^{K_H}$, the output of the layer is b where $b_i = sign(r_i)$. According to the chain rule we can obtain the gradient $\frac{\partial \mathcal{L}}{\partial r} = \frac{\partial \mathcal{L}}{\partial b}\frac{\partial b}{\partial r} = 0$, because $\frac{\partial b}{\partial r} = 0$. We adopt the following straight-through estimator to propagate the loss through the hashing layer instead.

$$\frac{\partial \mathcal{L}}{\partial r_i} = \begin{cases} \frac{\partial \mathcal{L}}{\partial b_i}, if -1 \le r_i \le 1 \\ 0, otherwise \end{cases} \tag{8}$$

Algorithm 1 Training algorithm with step size α.

Input: minibatch text encoding t, matching images x, mismatching images $\hat{x}$, number of training batch steps S
Output: generator G and discriminative hashing network H

procedure TRAIN(α, t, x, $\hat{x}$, S)
  Init(G);Init(H);
  for $i = 1$ to S **do**
    $z \sim \mathcal{N}(0,1)^Z$ {Draw sample of random noise}
    $\dot{x} \leftarrow G(z,t)$ {Forward through generator}
    Calculate $\mathcal{L}_H$ with $x, \dot{x}, \hat{x}, t$, according to Equ (6)
    $H \leftarrow H - \frac{\partial \mathcal{L}_H}{\partial H}$ {Update H}
    Calculate $\mathcal{L}_G$ with $\dot{x}, t$, according to Equ (7)
    $G \leftarrow G - \alpha\frac{\partial \mathcal{L}_G}{\partial G}$ {Update G}
  end for
  return G, H;
end procedure

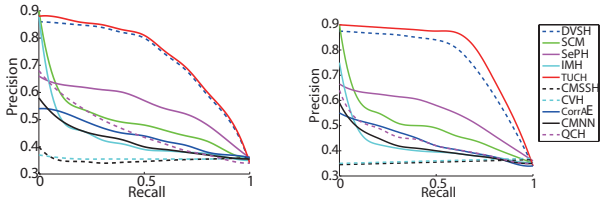
Moreover, when $|r_i| > 1$, we set the gradient to 0. To be clear, we consider $|r_i|$ as the confidence coefficient of $b_i = sign(r_i)$. If the confidence coefficient is large enough while training, we don't try to change this bit. Obviously, when we take output of the discrete hashing layer directly as the hash codes, the quantization error is 0. And the loss is computed directly by the binary codes, so that the loss can reflect fitting the degree of hashing codes on the training set well.

5 Experiments

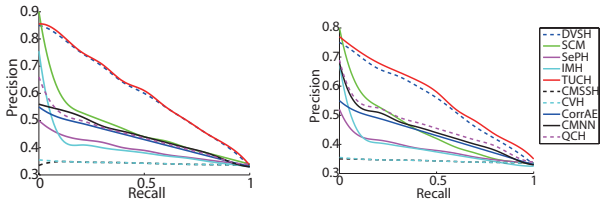
We conduct extensive experiments to evaluate the efficiency of the proposed TUCH model with several state-of-the-art hashing methods on two widely-used benchmark datasets.

5.1 Data Preparation

The evaluation is conducted on two benchmark cross-view datasets: **Microsoft COCO** [Lin *et al.*, 2014] and **IAPR TC-12** [Grubinger *et al.*, 2006].

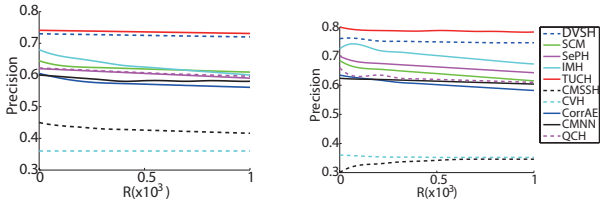


(a) $I \rightarrow T$ on COCO (b) $T \rightarrow I$ on COCO

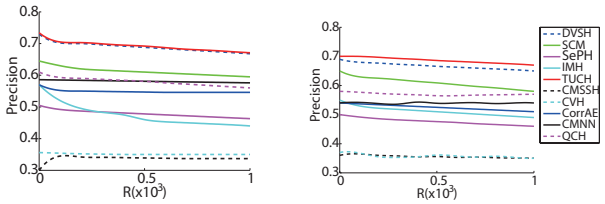


(c) $I \rightarrow T$ on IAPR TC-12 (d) $T \rightarrow I$ on IAPR TC-12

Figure 3: Precision-recall curves of cross-modal retrieval on Microsoft COCO and IAPR TC-12 @ 32 bits.



(a) $I \rightarrow T$ on COCO (b) $T \rightarrow I$ on COCO



(c) $I \rightarrow T$ on IAPR TC-12 (d) $T \rightarrow I$ on IAPR TC-12

Figure 4: Precision@top-R curves of cross-modal retrieval on Microsoft COCO and IAPR TC-12 @ 32 bits.

Microsoft COCO The current release of the recently proposed dataset contains 82783 training images and 40137 validation images. For each image, it provides at least five sentences annotations, belonging to 80 most frequent categories as ground truth labels. After pruning images with no category information, we get 82081 training images and randomly sample 5000 query images from the validation set along with their meta data.

IAPR TC-12 This dataset consists of 20000 images collected from a wide variety of domains, such as sports and actions, people, animals, cities, landscapes and so on. There are at least 1 sentence annotations for each image. Besides, it provides category annotations generated from segmentation tasks with 275 concepts. We prune the original IAPR TC-12 to form a new dataset, which consists of 18673 images with 22 most frequent concept tags.

5.2 Baseline and Evaluation Setup

We compare the cross-view retrieval performance of our approach with nine state-of-the-art cross-view hashing methods including three unsupervised methods **IMH** [Song *et al.*, 2013], **CVH** [Kumar and Udupa, 2011] and **CorrAE** [Feng *et al.*, 2014], and six supervised methods **CMSSH** [Bronstein *et al.*, 2010], **CM-NN** [Masci *et al.*, 2014], **SCM** [Zhang and Li, 2014], **QCH** [Wu *et al.*, 2015], **SePH** [Lin *et al.*, 2015] and **DVSH** [Cao *et al.*, 2016a], in which **CorrAE**,**CM-NN** and **DVSH** are deep methods.

We follow [Wu *et al.*, 2015; Zhang and Li, 2014; Cao *et al.*, 2016a] to evaluate the retrieval performance with Mean Average Precision (MAP), precision-recall curves, and precision@top-R curves. We adopt MAP@R=500 following the baseline method [Wu *et al.*, 2015; Cao *et al.*, 2016a]. And the sampled training set includes 5000 images along with all labels and sentences annotations for both **Microsoft COCO** and **IAPR TC-12**.

For **Microsoft COCO**, we randomly select 5000 images with annotations as training set for hashing, 1000 images with annotations as validation set and 1000 images with annotations as query set. For **IAPR TC-12**, we randomly select 5000 images with annotations as training set for hashing, 1000 images with annotations as validation set and 100 images with annotations per class as query set. The pairwise similarity labels for training are randomly constructed using semantic labels or concepts, and each pair is considered similar (dissimilar) if they share at least one (none) semantic label.

5.3 Training Details

We implement the TUCH model in the open-source **Torch7** framework. We take the text encoding model in [Reed *et al.*, 2016] to extract text encoding t. For training network, the generator G is a normal deconvolutional network as stated in [Reed *et al.*, 2016], and we employ the ResNet architecture as the image branch of the discriminative hashing network. We take pre-trained 18 layers ResNet as the initial value of the image branch to fine-tune.

For both datasets, we first train the unsupervised text encoding model with all the sentences descriptions except for query set. And then we train an unsupervised GAN model with G and discriminative sub-network of H. The training data is all the sentence-image pairs without tag except for query set. Finally, we add the hashing layer and the semantic embedding loss into H along with the discriminative loss, and perform a supervised training on small labeled training set. The TUCH approach involves 4 penalty parameters λ_1, λ_2, λ_H and λ_G for trading off the relative importance of irrelevant image, fake image, discriminative loss for H and discriminative loss for G in Equ (6) and Equ (7). And we can achieve good results with $\lambda_1 = \lambda_2 = 0.5$ and $\lambda_H = \lambda_G = 0.01$.

5.4 Results and Discussions

We compare our approach TUCH with the nine baseline methods on the two datasets in term of MAP, precision-recall curve and precision@top-R curves of two cross-view retrieval

Table 1: Mean Average Precision (MAP) Comparison of Cross-View Retrieval Tasks on Two Datasets

Task	Method	Microsoft COCO				IAPR TC-12			
		16bit	32bit	64bit	128bit	16bit	32bit	64bit	128bit
I → T	CMSSH [Bronstein *et al.*, 2010]	0.4047	0.4886	0.4405	0.4480	0.3445	0.3371	0.3478	0.3738
	CVH [Kumar and Udupa, 2011]	0.3731	0.3677	0.3657	0.3570	0.3788	0.3686	0.3620	0.3540
	IMH [Song *et al.*, 2013]	0.6154	0.6505	0.6573	0.6770	0.4632	0.4901	0.5104	0.5212
	CorrAE [Feng *et al.*, 2014]	0.5498	0.5559	0.5695	0.5809	0.4951	0.5252	0.5578	0.5890
	CM-NN [Masci *et al.*, 2014]	0.5557	0.5602	0.5847	0.5938	0.5159	0.5419	0.5766	0.6003
	SCM [Zhang and Li, 2014]	0.5699	0.6002	0.6307	0.6487	0.5880	0.6110	0.6282	0.6370
	QCH [Wu *et al.*, 2015]	0.5723	0.5954	0.6132	0.6345	0.5259	0.5546	0.5785	0.6054
	SePH [Lin *et al.*, 2015]	0.5813	0.6134	0.6253	0.6339	0.5070	0.5130	0.5151	0.5309
	DVSH [Cao *et al.*, 2016a]	0.5870	0.7132	**0.7386**	0.7552	0.5696	0.6321	**0.6964**	**0.7236**
	TUCH	**0.6280**	**0.7135**	0.7349	**0.7660**	**0.5953**	**0.6372**	0.6902	0.7144
T → I	CMSSH [Bronstein *et al.*, 2010]	0.3747	0.3838	0.3400	0.3601	0.3633	0.3770	0.3645	0.3482
	CVH [Kumar and Udupa, 2011]	0.3734	0.3686	0.3645	0.3711	0.3790	0.3674	0.3636	0.3560
	IMH [Song *et al.*, 2013]	0.6068	0.6793	0.7280	0.7403	0.5157	0.5259	0.5337	0.5274
	CorrAE [Feng *et al.*, 2014]	0.5593	0.5807	0.6109	0.6262	0.4975	0.5195	0.5329	0.5495
	CM-NN [Masci *et al.*, 2014]	0.5793	0.5984	0.6195	0.6448	0.5119	0.5394	0.5487	0.5649
	SCM [Zhang and Li, 2014]	0.5581	0.6188	0.6583	0.6858	0.5876	0.6045	0.6200	0.6262
	QCH [Wu *et al.*, 2015]	0.5742	0.6057	0.6375	0.6669	0.4997	0.5364	0.5652	0.5885
	SePH [Lin *et al.*, 2015]	0.6127	0.6496	0.6723	0.6929	0.4712	0.4801	0.4812	0.4955
	DVSH [Cao *et al.*, 2016a]	0.5906	0.7365	0.7583	0.7673	0.6037	0.6395	0.6806	0.6751
	TUCH	**0.6493**	**0.7602**	**0.7864**	**0.8117**	**0.6239**	**0.6563**	**0.6876**	**0.7088**

tasks: image query on sentence database ($I \rightarrow T$), and sentence query on image database ($T \rightarrow I$).

We evaluate all methods with 16, 32, 64 and 128 bits hash codes. The results are reported in Table 1. From the experimental results, we can observe that TUCH nearly outperforms all baseline methods for most cross-view tasks on the benchmark datasets. Specifically, compared to the state-of-the-art sentence-image hashing retrieval baseline **DVSH**, TUCH achieves absolute increases of 1.1%/3.9% and 0.4%/1.9% in average MAP of different code length for two cross-view retrieval tasks $I \rightarrow T$ and $T \rightarrow I$ on Microsoft COCO and IAPR TC-12 datasets. The performance of unsupervised baseline methods differs on the two benchmark datasets. Compared to the best unsupervised method **IMH** on Microsoft COCO, TUCH get absolute increases of 5.6% and 6.3% on $I \rightarrow T$ and $T \rightarrow I$ tasks, while the data on IAPR TC-12 is 11.8% and 14.4% which is compared with **CorrAE**. We can observe that TUCH gains more increases in the $T \rightarrow I$ task on both benchmark datasets. It is because the hashing function is directly over image domain, and the information of image is reserved as much as possible, so that hash codes of images shows good aggregation characteristics. Once the hash codes of texts are place in the right semantic position, the relevant images can be found easily.

The precision-recall curves with 32 bits for the two cross-view tasks $I \rightarrow T$ and $T \rightarrow I$ on two datasets Microsoft COCO and IAPR TC-12 are shown in Figure 3, respectively. TUCH shows the best cross-view retrieval performance at all recall levels on $T \rightarrow I$ task, and almost best performance on $I \rightarrow T$ task. Figure 4 shows the precision@top-R curves of all comparison methods with 32 bits on the two datasets, which shows how the precision changes with the number R of top-retrieved results. TUCH outperforms other baseline method again and shows effectiveness and stability on different levels of R value.

6 Conclusion

This paper presents a novel deep model that is able to turn cross-view hashing into single-view hashing on image domain via GANs (TUCH) for sentence-image cross-view hashing, thus enabling the information of image to be preserved as much as possible. Our TUCH model converts sentences into images and then solves the multi-view hashing problem with a multi-task and end-to-end generative adversarial learning architecture. Besides, we add a discrete hashing layer in TUCH which directly outputs binary codes via loss computation, which avoids the information leak during the feature quantization. Comprehensive empirical evidence shows that our TUCH approach achieves state-of-the-art results, especially on text to image retrieval task, on image-sentences datasets, i.e. standard IAPRTC-12 and large-scale Microsoft COCO. In the future, we plan to extend TUCH on mobile computing, which relies on good model compression approach and parallel computing algorithm with high speed on mobile device.

References

[Bronstein *et al.*, 2010] Michael M. Bronstein, Alexander M. Bronstein, Fabrice Michel, and Nikos Paragios. Data fusion through cross-modality metric learning using similarity-sensitive hashing. In *CVPR*, 2010.

[Cao *et al.*, 2016a] Yue Cao, Mingsheng Long, Jianmin Wang, Qiang Yang, and Philip S. Yu. Deep visual-semantic hashing for cross-modal retrieval. In *KDD*, 2016.

[Cao *et al.*, 2016b] Yue Cao, Mingsheng Long, Jianmin Wang, Han Zhu, and Qingfu Wen. Deep quantization network for efficient image retrieval. In *AAAI*, 2016.

[Denton *et al.*, 2015] Emily L. Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*, 2015.

[Ding *et al.*, 2016] Guiguang Ding, Yuchen Guo, Jile Zhou, and Yue Gao. Large-scale cross-modality search via collective matrix factorization hashing. *TIP*, pages 5427–5440, 2016.

[Feng *et al.*, 2014] Fangxiang Feng, Xiaojie Wang, and Ruifan Li. Cross-modal retrieval with correspondence autoencoder. In *Multimedia*, 2014.

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

[Gong *et al.*, 2013] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *PAMI*, 2013.

[Goodfellow *et al.*, 2014] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[Grubinger *et al.*, 2006] Michael Grubinger, Paul Clough, Henning Müller, and Thomas Deselaers. The iapr tc-12 benchmark: A new evaluation resource for visual information systems. In *International workshop ontoImage*, volume 5, 2006.

[Guo *et al.*, 2016] Yuchen Guo, Guiguang Ding, Jungong Han, and Xiaoming Jin. Robust iterative quantization for efficient ℓ_p-norm similarity search. In *IJCAI2016*, pages 3382–3388, 2016.

[He *et al.*, 2013] Kaiming He, Fang Wen, and Jian Sun. K-means hashing: An affinity-preserving quantization method for learning binary compact codes. In *CVPR*, 2013.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[Kong and Li, 2012] Weihao Kong and Wu-Jun Li. Isotropic hashing. In *NIPS*, 2012.

[Kumar and Udupa, 2011] Shaishav Kumar and Raghavendra Udupa. Learning hash functions for cross-view similarity search. In *IJCAI*, 2011.

[Lai *et al.*, 2015] Hanjiang Lai, Yan Pan, Ye Liu, and Shuicheng Yan. Simultaneous feature learning and hash coding with deep neural networks. In *CVPR*, 2015.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, 2014.

[Lin *et al.*, 2015] Zijia Lin, Guiguang Ding, Mingqing Hu, and Jianmin Wang. Semantics-preserving hashing for cross-view retrieval. In *CVPR*, 2015.

[Lin *et al.*, 2016] Zijia Lin, Guiguang Ding, Jungong Han, and Jianmin Wang. Cross-view retrieval via probability-based semantics-preserving hashing. *Transactions on Cybernetics*, 2016.

[Liu and Lu, 2016] Shicong Liu and Hongtao Lu. Accurate deep representation quantization with gradient snapping layer for similarity search. *CoRR*, 2016.

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

[Liu *et al.*, 2014] Wei Liu, Cun Mu, Sanjiv Kumar, and Shih-Fu Chang. Discrete graph hashing. In *NIPS*, 2014.

[Masci *et al.*, 2014] Jonathan Masci, Michael M. Bronstein, Alexander M. Bronstein, and Jürgen Schmidhuber. Multimodal similarity-preserving hashing. *PAMI*, 2014.

[Radford *et al.*, 2015] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, 2015.

[Reed *et al.*, 2016] Scott E. Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *ICML*, 2016.

[Shen *et al.*, 2015a] Fumin Shen, Chunhua Shen, Wei Liu, and Heng Tao Shen. Supervised discrete hashing. In *CVPR*, 2015.

[Shen *et al.*, 2015b] Fumin Shen, Chunhua Shen, Qinfeng Shi, Anton van den Hengel, Zhenmin Tang, and Heng Tao Shen. Hashing on nonlinear manifolds. *TIP*, 2015.

[Smeulders *et al.*, 2000] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh C. Jain. Content-based image retrieval at the end of the early years. *PAMI*, 2000.

[Song *et al.*, 2013] Jingkuan Song, Yang Yang, Yi Yang, Zi Huang, and Heng Tao Shen. Inter-media hashing for large-scale retrieval from heterogeneous data sources. In *SIGMOD*, 2013.

[Wang *et al.*, 2016] Jun Wang, Wei Liu, Sanjiv Kumar, and Shih-Fu Chang. Learning to hash for indexing big data - A survey. *Proceedings of the IEEE*, 2016.

[Wei *et al.*, 2014] Ying Wei, Yangqiu Song, Yi Zhen, Bo Liu, and Qiang Yang. Scalable heterogeneous translated hashing. In *KDD*, 2014.

[Weiss *et al.*, 2008] Yair Weiss, Antonio Torralba, and Robert Fergus. Spectral hashing. In *NIPS*, 2008.

[Wu *et al.*, 2015] Botong Wu, Qiang Yang, Wei-Shi Zheng, Yizhou Wang, and Jingdong Wang. Quantized correlation hashing for fast cross-modal search. In *IJCAI*, 2015.

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

[Xu *et al.*, 2013] Bin Xu, Jiajun Bu, Yue Lin, Chun Chen, Xiaofei He, and Deng Cai. Harmonious hashing. In *IJCAI*, 2013.

[Zhang and Li, 2014] Dongqing Zhang and Wu-Jun Li. Large-scale supervised multimodal hashing with semantic correlation maximization. In *AAAI*, 2014.

[Zhang *et al.*, 2014] Peichao Zhang, Wei Zhang, Wu-Jun Li, and Minyi Guo. Supervised hashing with latent factor models. In *SIGIR*, 2014.

[Zhen and Yeung, 2012] Yi Zhen and Dit-Yan Yeung. Co-regularized hashing for multimodal data. In *NIPS*, 2012.

[Zhu *et al.*, 2016] Han Zhu, Mingsheng Long, Jianmin Wang, and Yue Cao. Deep hashing network for efficient similarity retrieval. In *AAAI*, 2016.