SitNet: Discrete Similarity Transfer Network for Zero-shot Hashing*

Yuchen Guo†, Guiguang Ding†, Jungong Han‡, Yue Gao†
†School of Software, Tsinghua University, Beijing 100084, China
‡School of Computing & Communications, Lancaster University, UK
yuchen.w.guo@gmail.com, {dinggg,gaoyue}@tsinghua.edu.cn,jungong.han@northumbria.ac.uk

Abstract

Hashing has been widely utilized for fast image retrieval recently. With semantic information as supervision, hashing approaches perform much better, especially when combined with deep convolution neural network(CNN). However, in practice, new concepts emerge every day, making collecting supervised information for re-training hashing model infeasible. In this paper, we propose a novel **zero-shot** hashing approach, called Discrete **Si**milarity **T**ransfer **Net**work (SitNet), to preserve the semantic similarity between images from both "seen" concepts and new "unseen" concepts. Motivated by zero-shot learning, the semantic vectors of concepts are adopted to capture the similarity structures among classes, making the model trained with seen concepts generalize well for unseen ones. We adopt a multi-task architecture to exploit the supervised information for seen concepts and the semantic vectors simultaneously. Moreover, a discrete hashing layer is integrated into the network for hashcode generating to avoid the information loss caused by real-value relaxation in training phase, which is a critical problem in existing works. Experiments on three benchmarks validate the superiority of SitNet to the state-of-the-arts.

1 Introduction

The recent decade has witnessed the fast development of hashing for semantic image retrieval [Wang *et al.*, 2016]. By binarizing real-valued image feature vectors into '0/1' bit sequences, hashing can index large-scale image database with small storage cost and enable efficient similarity search based on the bit-wise XOR operation. Starting from data-independent approaches [Gionis *et al.*, 1999] which utilizes no prior knowledge about data, recent works mostly focus on data-dependent hashing, which leverages information inside data itself. There are two main streams, unsupervised hashing like Iterative Quantization [Gong *et al.*, 2013] and supervised hashing like Supervised Discrete Hashing [Shen *et al.*, 2015a]. With the supervised information like semantic similarity matrix or class labels, the supervised approaches achieve superior retrieval performance because the intrinsic semantic property in the data is better explored.

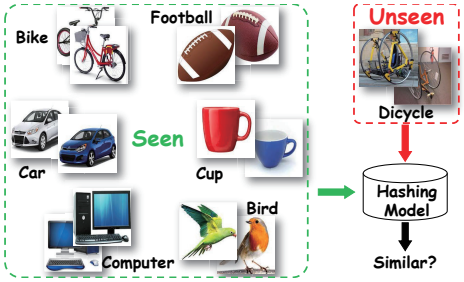


Figure 1: Zero-shot hashing. The hashing model trained with seen concepts should generalize well on the unseen concepts.

Recently the deep convolutional neural network (CNN) has achieved great success in many computer vision tasks, like image classification[He *et al.*, 2016] and face recognition[Wen *et al.*, 2016]. Inspired by CNN's powerful feature extraction ability, some works have attempted to build hashing models based on CNN [Lai *et al.*, 2015; Liu *et al.*, 2016; Xia *et al.*, 2014] have appeared. They require the hashcodes produced by the last fully connected layer to preserve the similarity given by the supervised information. It is demonstrated that the image retrieval accuracy is significantly improved by CNN-based hashing approaches compared with the non-CNN ones [Liu *et al.*, 2016].

It should be noticed that the existing hashing approaches mainly focus on the close-set retrieval, i.e., the concepts of possible testing samples (both database samples and query samples) are within the training set. However, the explosive growth of Web images violates this setting because the new concepts about the images may emerge rapidly. It is expensive to annotate sufficient training data for the new concepts timely, and also, impractical to retrain the hashing model whereas the retrieval system meets a new concept. As illustrated in Figure 1, the existing approaches perform well on the seen concepts because they are given correct guidance, but they may easily fail on the unseen concepts that they never meet before such as the "dicycle" which is a kind of vehicle

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

with two wheels side by side. Hence, building generalizable hashing model that can produce effective hashcodes for unseen concepts is very important for real-world applications. However, existing works pay little attention to this problem.

In this paper, we consider the zero-shot hashing (ZSH) problem [Yang *et al.*, 2016] which aims to build hashing model that can capture the similarity structure of both seen and unseen concepts. One challenge in ZSH is how to deduce the information of unseen concepts from the seen concepts. In fact, one important reason why existing works fail to handle unseen concepts is that they treat all concepts independently such that each concept cannot leverage the knowledge from other related concepts or contribute its own information to the others. Fortunately, recent progress of zero-shot classification [Changpinyo *et al.*, 2016; Guo *et al.*, 2017a] shows that the relationship between seen and unseen concepts can be well characterized in the word embedding space [Turian *et al.*, 2010] and word vectors of classes can be utilized as an effective tool to transfer knowledge among classes. Inspired by this idea, we propose a novel ZSH approach based on CNN, termed as Discrete Similarity Transfer Network (SitNet), which produces semantic-similarity-preserving hashcodes for both seen and unseen concepts. Specifically, we consider three important aspects. The first is similarity transfer. We utilize the semantic vectors of concepts as the side information and enforce the hashcodes produced by the network to capture the semantic structure in the word embedding space. Due to the transferability of the space, the model trained with seen concepts can also produce hashcodes capturing the characteristics of unseen concepts [Socher *et al.*, 2013]. In this way, the generalization ability of the hashing model is improved. The second is discriminability. To achieve this goal, we adopt a regularized center loss [Wen *et al.*, 2016] alongside a semantic vector guided loss. The above two aspects are jointly optimized in a multi-task architecture. Thirdly, different from many previous CNN hashing works which utilize a disjoint strategy for hashcodes generation and hashing function learning [Xia *et al.*, 2014], or adopt a real-value relaxation in the network [Liu *et al.*, 2016], we adopt a discrete layer in the network that directly generates binary codes. With the discrete layer, information leak is prevented during quantization. Besides, we adopt a simple and efficient method to backpropagate the loss through the discrete layer. In summary, we make the following contributions in this paper:

- We put forward a novel approach termed as SitNet for zero-shot hashing, which is capable of producing effective hashcodes for samples from unseen concepts by transferring knowledge via the word vector space.
- A multi-task architecture is adopted simultaneously considering the supervised information from the seen concepts and relationship among concepts such that limited seen knowledge can help build effective hashing model for unseen concepts. To our best knowledge, this is the first CNN hashing work in the zero-shot way.
- We design a discrete layer which directly outputs binary codes for loss computation, avoiding the information loss during quantization. An efficient algorithm is adopted for loss backpropagation for the discrete layer.

2 Related Work

2.1 Hashing for Retrieval

Hashing is a widely used indexing technique for image retrieval. Locality Sensitive Hashing [Gionis *et al.*, 1999] is the seminal hashing work. As it is data independent, it usually requires long hashcodes for satisfactory performance. Therefore, many data-dependent hashing approaches are proposed, which fall into two categories: unsupervised and supervised hashing. Regarding unsupervised hashing, the unsupervised information of data is considered, like the manifold structure [Guo *et al.*, 2017b; Shen *et al.*, 2015b], and the variance of feature[Gong *et al.*, 2013; Guo *et al.*, 2016]. Supervised hashing uses the supervised knowledge to capture the semantic property of data, like Supervised Hashing with Kernels [Liu *et al.*, 2012] and Supervised Discrete Hashing [Shen *et al.*, 2015a]. As more information is used, supervised approaches achieve better results than the unsupervised ones [Ding *et al.*, 2016; Lin *et al.*, 2016].

Recently, researchers have attempted to combine the deep learning with hashing. Xia *et al.* [2014] propose a disjoint strategy which firstly generates hashcodes using the supervised knowledge and then utilizes CNN to construct the hashing functions. Alternatively, Lai *et al.* [2015] propose an end-to-end network to minimize the triplet ranking loss. Liu *et al.* [2016] and Zhu *et al.* [2016] propose simultaneously minimizing the pair-wise similarity loss and the quantization loss. Benefiting from the power of CNN, they achieve significant improvement over the traditional approaches. However, the aforementioned approaches focus on close-set retrieval, leading to poor performance on the unseen concepts which is a common situation in Web based retrieval tasks. Yang *et al.* [2016] noticed this problem and propose the zero-shot hashing schema which significantly improves the generalization ability of the hashing model trained with seen concepts. However, they fail to effectively exploit the supervised information and leverage the power of deep learning. Please refer to [Wang *et al.*, 2016] for more elaborate survey on hashing.

2.2 Zero-shot Learning

Zero-shot learning (ZSL) is to construct models for concepts with no training data [Lampert *et al.*, 2014]. Because no data for unseen concepts is available, ZSL utilizes the shared attribute space as intermediate to transfer supervised knowledge from seen concepts to unseen concepts. Farhadi *et al.* [2009] and Lampert *et al.* [2014] proposed two seminal works of attribute-based ZSL. Unfortunately, the human-defined attributes are noisy and expensive to obtain, more reliable and scalable representation for concepts are desired. So Socher *et al.* [2013] used the word embedding of concepts for knowledge transfer. Motivated by [Socher *et al.*, 2013], many follow-up works are proposed recently [Akata *et al.*, 2016; Changpinyo *et al.*, 2016; Guo *et al.*, 2017a]. They demonstrate that the word vector space can well capture the relationship between seen and unseen concepts and is helpful for knowledge transfer. Therefore, their models trained with only seen concepts generalize well on unseen concepts. However, existing works mostly focus on classification task, and how to combine them with hashing is still an open problem.

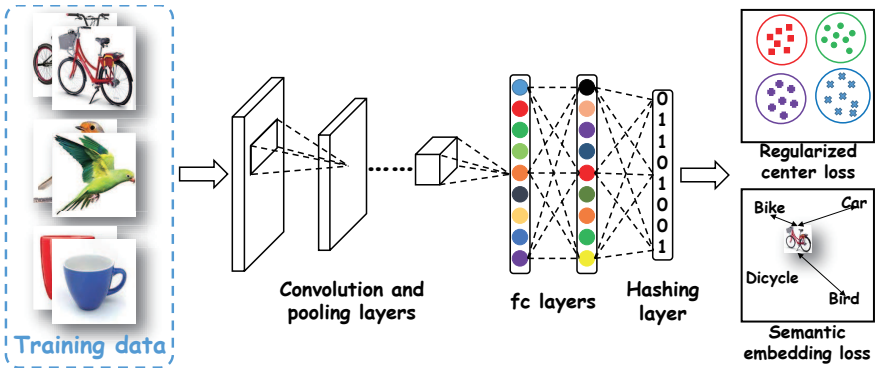


Figure 2: Architecture overview of SitNet. Each training image belongs to a concept/class and each concept has a semantic embedding vector from word2vec. We incorporate a discrete hashing layer to generate hashcodes. Based on the hashcodes, we compute the regularized center loss, and the semantic embedding loss to enforce the hashcodes to capture the similarity relationship among concepts. Because of the transferability of the semantic vector space, the model can generalize well for the unseen concepts.

3 Discrete Similarity Transfer Network

3.1 Problem Definition

The definition of ZSH in this paper follows [Yang *et al.*, 2016]. We are given n training images $\{I_1, ..., I_n\}$ and each image I_m belongs to one visual concept from set $\mathcal{C}^s = \{c_1^s, ..., c_{k_s}^s\}$ where k_s is the size of $\mathcal{C}^s$. For each image I_m, there is a label vector $y_m \in \{0,1\}^{k_s}$ where $y_{mj} = 1$ if I_m belongs to concept c_j^s and $y_{mj} = 0$ otherwise. In the conventional hashing setting, it is assumed that the training data and testing data are from the seen set $\mathcal{C}^s$. In the ZSH setting, we assume that some testing samples are from an unseen concept set $\mathcal{C}^u = \{c_1^u, ..., c_{k_u}^u\}$ with $\mathcal{C}^s \cap \mathcal{C}^u = \emptyset$ where k_u is the size of $\mathcal{C}^u$. By using training samples only from $\mathcal{C}^s$ where no samples of $\mathcal{C}^u$ is available, we aim to learn a hashing model $h : I \mapsto \{-1,1\}^H$ to map image to H-bit hashcodes. It is desired that h can preserve the semantic similarity between samples from both $\mathcal{C}^s$ and $\mathcal{C}^u$ in the Hamming space even if there is no data from $\mathcal{C}^u$ available during the training phase. Furthermore, to transfer knowledge across concepts, each concept $c \in \mathcal{C}^s \cup \mathcal{C}^u$ has a 300-dimensional semantic vector obtained from word2vec, denoted as $\mathbf{v}_c$.

3.2 Network Architecture

The architecture of SitNet is illustrated in Figure 2. The convolution, pooling, and fully-connected layers follow some well-established architectures considering their effectiveness in many tasks. We adopt AlexNet [Krizhevsky *et al.*, 2012] in this paper. We incorporate a discrete hashing laye fully connected to the last fc layer and has H output units where we use the sign function as the activation function for discretization. It can directly output the hashcode of a sample, which is clearly different from existing end-to-end CNN hashing approaches that need real-value relaxation during training [Shen *et al.*, 2015a]. Next, another fully-connected layer is adopted to project the hashcodes into the semantic vector space, in which the loss is computed. Our loss function consists of two parts, where the first is a max-margin loss between the projected representation and the target semantic vector $\mathbf{v}_c$ to preserve the similarity structure among all concepts for knowledge transfer. The second is a regularized center loss which leverages the supervised information from the seen data.

3.3 Loss Function

Given an input image I_m, suppose the output of the hashing layer is $\mathbf{b}_m \in \{-1,1\}^H$ is the H-bit hashcodes. Then the hashcodes are projected to the final semantic vector space by the parameter $\mathbf{W}^\ell$ and the output is denoted as $\mathbf{x}_m$.

As demonstrated by the achievement of zero-shot classification approaches [Lampert *et al.*, 2014; Socher *et al.*, 2013], the semantic vector space is an effective tool for knowledge transfer across concepts. A model trained with the seen concepts that maps images to the semantic vector space can also work well on the unseen concepts, i.e., it can map a testing image from an unseen concept to the word vector of its concept's even though they are not observed before. Motivated by this idea, we also expect that hashcode can capture the similarity structure in this space. Formally, this idea can be implemented as $\mathbf{b}_m\mathbf{W}^\ell = \mathbf{x}_m \approx \mathbf{v}_{c_{I_m}}$, where c_{I_m} is the concept that I_m belongs to. For trainingk, the loss is designed as:

$$\mathcal{L}_e = \sum\nolimits_m \max(0, \lambda + \|\mathbf{x}_m - \mathbf{v}_{c_{I_m}}\|^2 - \min_{c' \neq c_{I_m}} \|\mathbf{x}_m - \mathbf{v}_{c'}\|^2) \quad (1)$$

where λ is a margin parameter. We adopt the max-margin loss as it leads to more generalizable model and it can address the hubness problem [Lazaridou *et al.*, 2015] in zero-shot learning while some other loss functions, like ridge loss, fail to do so. By the semantic embedding loss, the learned hashing model can capture the relationship among concepts and thus it works better on the unseen concepts.

Unlike existing hashing approaches which treat all concepts independently, our approach can quantitatively measure and capture the semantic correlations among concepts in the semantic vector space. In this way, the supervised information from one concept can contribute to other concepts. For example, the concept "dicycle" is not observed by the hashing model during training. But its word vector is close to the vectors of "bicycle" and "car". Therefore, the hashing model can obtain some valuable knowledge from "bicycle" and "car" to deduce what hashcode a "dicycle" image should have. So,

given a "dicycle" image, the hashing model can correctly produce its hashcode that is near but different from "bicycle" or "car", and far from unrelated concepts like "dog".

The semantic embedding loss mostly focuses on improving the generalizability of the hashing model. Meanwhile, it is also desired that the samples from the same concept have very similar hashcodes whereas samples from different concepts have dissimilar hashcodes. To preserve similarity, we adopt a regularized center loss to learn discriminative hashcodes [Wen *et al.*, 2016]. Specifically, for the k_s seen concepts, there are k_s centers $\{\mathbf{t}_1, ..., \mathbf{t}_{k_s}\}$ in the semantic vector space corresponding to them. To achieve our goal, we propose the following regularized center loss:

$$\begin{aligned}\mathcal{L}_c = &-\sum_m \log \frac{\exp(\mathbf{x}_m \mathbf{v}'_{c_{I_m}})}{\sum_{c=1}^{k_s} \exp(\mathbf{x}_m \mathbf{v}'_c)} \\ &+ \frac{\alpha}{2} \sum_m \|\mathbf{x}_m - \mathbf{t}_{c_{I_m}}\|^2 - \frac{\beta}{2} \sum_c \min_{c' \neq c} \|\mathbf{t}_c - \mathbf{t}_{c'}\|^2\end{aligned} \tag{2}$$

where α and β are the weight parameters. The first term is a semantic vector guided soft-max loss that adopts the (normalized) semantic vector of each concept as the basis. The reason why we fix the parameters to the semantic vectors is that our goal is to adjust $\mathbf{x}_m$ (intrinsically equivalent to the hashcodes) to improve the separability between concepts instead of seeking parameters to separate them. The second term enables to cluster the samples from the same concept [Wen *et al.*, 2016] such that they have similar hashcodes. The third term is a discriminability regularization to push the centers far away to each other. The latter two terms can improve the discriminability of the hashcodes. In summary, minimizing the regularized center loss can adjust the hashcodes to achieve both high intra-concept similarity and inter-concept distance.

For the conventional retrieval task, the regularized center loss is able to result in good performance. However, in the zero-shot scenario, it is necessary to consider the semantic embedding loss. Therefore, we adopt a multi-task architecture to jointly minimize these two losses, which can be implemented as minimizing the following weighted loss:

$$\mathcal{L} = \mathcal{L}_c + \gamma \mathcal{L}_e \tag{3}$$

3.4 Optimization

We adopt the backpropagation algorithm with mini-batch stochastic gradient descent method to train the network ande just need the following gradients to backpropagate the loss:

$$\frac{\partial \mathcal{L}_e}{\partial \mathbf{x}_m} = \begin{cases} 0, \text{if} \|\mathbf{x}_m - \mathbf{v}_{c'}\|^2 - \|\mathbf{x}_m - \mathbf{v}_{c_{I_m}}\|^2 \geq \lambda \\ \mathbf{v}_{c'} - \mathbf{v}_{c_{I_m}}, \text{ otherwise} \end{cases} \tag{4}$$

$$\frac{\partial \mathcal{L}_c}{\partial \mathbf{x}_m} = \frac{\partial \mathcal{L}_s}{\mathbf{x}_m} + \alpha(\mathbf{x}_m - \mathbf{t}_{c_{I_m}}) \tag{5}$$

where $c' = \text{argmin}_{c \neq c_{I_m}} \|\mathbf{x}_m - \mathbf{v}_c\|^2$ and $\mathcal{L}_s$ denotes the traditional soft-max loss. In some well-established deep learning tools, like Caffe [Jia *et al.*, 2014], they just need the above gradients and their build-in operations can backpropagate the loss through the network to minimize the training set loss.

However, there is still a problem during backpropagation caused by the hashing layer because its 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" [Bengio *et al.*, 2013] to compute the gradients. Specifically, for training sample I_m, suppose its pre-activation/discretization representation for the hashing layer is $\mathbf{r}_m \in \mathbb{R}^H$ and its hashcode is $\mathbf{b}_m$ where $b_{mi} = \text{sign}(r_{mi})$. Based on the chain rule, we can obtain the gradient $\frac{\partial \mathcal{L}}{\partial b_{mi}}$. Obviously, in the conventional way, we have $\frac{\partial \mathcal{L}}{\partial r_{mi}} = \frac{\partial \mathcal{L}}{\partial b_{mi}} \frac{\partial b_{mi}}{r_{mi}} = 0$ because $\frac{\partial \text{sign}(r_{mi})}{\partial r_{mi}} = 0$. Instead, we propose to adopt the following straight-through estimator to propagate the loss through the hashing layer:

$$\frac{\partial \mathcal{L}}{\partial r_{mi}} = \begin{cases} \frac{\partial \mathcal{L}}{\partial b_{mi}}, \text{ if } -1 \leq r_{mi} \leq 1 \\ 0, \text{ otherwise} \end{cases} \tag{6}$$

In fact, the gradient in Eq. (6) is reasonable in hashing. For example, when b_{mi} has a wrong value and it should be changed (say, $b_{mi} = 1$ but it should be -1), $\frac{\partial \mathcal{L}}{\partial b_{mi}}$ will push b_{mi} to the negative side. Because we have $b_{mi} = \text{sign}(r_{mi})$ which indicates b_{mi} and r_{mi} have the same sign, we also need to push r_{mi} to the negative side such that its sign is turned over. Consequently, the direction to which the loss pushes b_{mi} is the same as the direction to which the loss pushes r_{mi}. Therefore, it is a reasonable choice to directly pass $\frac{\partial \mathcal{L}}{\partial b_{mi}}$ through the sign function to $\frac{\partial \mathcal{L}}{\partial r_{mi}}$. Moreover, when $|r_{mi}| > 1$ we set the gradient to 0. In fact, we can regard r_{mi} as the confidence that b_{mi} takes 1 or -1. When $|r_{mi}|$ is small, it is more likely that b_{mi} takes the wrong sign such that it is necessary to adjust them. But when $|r_{mi}|$ is large, this bit is reasonably trustable. In fact, the network is influenced by the whole training set. During the training process, the loss caused by one bit changes even if it is fixed since the network is tuned by other training samples. Therefore, we ignore it and focus on the bits with low confidence. By progressively assigning more correct hashcodes, the loss of one bit will decrease with training process. Finally, the loss caused by the high-confidence bits is usually small in a well-trained network.

The existing CNN hashing approaches adopt two strategies to account for the discrete optimization. The first strategy is the disjoint optimization [Xia *et al.*, 2014] which firstly generates the binary codes using the supervised information and then train a network to map samples to the hashcodes as a multi-bit binary classification problem. This strategy usually gives rise to sub-optimal results [Lai *et al.*, 2015]. The second strategy is real-value relaxation. This strategy is adopted by the end-to-end architecture which relaxes the sign function to the sigmoid function [Lai *et al.*, 2015], identity function [Liu *et al.*, 2016], or some other differentiable functions. In addition, a quantization loss is always integrated into the loss function in order to enforce the output to be close to 1 or -1. However, because they treat the quantization loss as a weighted part of loss function, the network may try to minimize the quantization loss such that its primary objective is not fully optimized. Moreover, their loss is computed by the relaxed real-value outputs. Hence, the obtained network is not optimal for binary codes. But in SitNet, we directly adopt a discrete hashing layer, thereby only focusing on the primary objective function in Eq. (3). In addition, because the op-

timization is directly performed to the binary codes without any relaxation, our approach can lead to the optimal network for binary codes.

Finally, to update the center $\mathbf{t}_c$ for each concept, we can utilize a mini-batched based updating rule [Wen *et al.*, 2016]:

$$\Delta \mathbf{t}_c = \frac{\sum_m \delta(c_{I_m} = c) \cdot (\mathbf{t}_c - \mathbf{x}_m)}{1 + \sum_m \delta(c_{I_m} = c)} - \frac{\beta}{\alpha}(\mathbf{t}_c - \mathbf{t}_{c'}) \quad (7)$$

where $\delta(x)$ is an indicator function which is 1 when x is true or 0 otherwise, and $c' = \text{argmin}_{c'} \|\mathbf{t}_c - \mathbf{t}_{c'}\|^2$. The center is updated by $\mathbf{t}_c \leftarrow \mathbf{t}_c - \tau \Delta \mathbf{t}_c$ where τ denotes a tiny stepsize.

4 Experiment

4.1 Data Preparation

Animals with Attributes [Lampert *et al.*, 2014]. AwA dataset consists of $30,475$ images manually labeled by 50 animal categories, such as "tiger", "dolphin", and etc. This is a widely used dataset for zero-shot learning.

CIFAR10 [Krizhevsky *et al.*, 2012]. CIFAR10 dataset contains 10 non-overlapping objects like "cat" and "truck" and each object has $6,000$ images with 32×32 size. It is frequently utilized for evaluating the hashing approaches.

ImageNet [Deng *et al.*, 2009]. ImageNet is a large-scale vision dataset organized according to WordNet hierarchy. In our experiment, we use the subset of ImageNet for the Large Scale Visual Recognition Challenge 2012 which has over 1.2m images manually labeled by $1,000$ concepts.

Semantic Vector. Semantic vector is an important part in the proposed approach for knowledge transfer across concepts to facilitate zero-shot hashing. In our experiment, we use the word2vec tool. Specifically, we use the name of the concept as the input for word2vec and adopt its 300-dimensional output as the semantic vector for the concept.

4.2 Settings

We adopt the benchmark datasets to construct the zero-shot scenario following [Yang *et al.*, 2016]. For AwA dataset, we split the categories in to 5 groups where each group has 10 categories. We use one group as the unseen concepts and the other four as seen concepts and thus we have 5 different seen-unseen splits. For CIFAR10 dataset, we use one category as unseen and the other nine as seen categories which leads to 10 seen-unseen splits. For ImageNet dataset, following [Yang *et al.*, 2016], we randomly select 100 categories which have the semantic vector from word2vec which gives us about $130,000$ images. We use 10 categories as unseen and the other 90 as seen. For all three datasets, we construct the training and query set as follows. From the seen concepts, we randomly select $10,000$ images as the training set. For testing, we randomly select $1,000$ images from the unseen concepts as the query set. The remaining unseen category images and all seen category images form the retrieval database. Obviously, this setting is different from the conventional one in the existing hashing approaches where the concepts in the testing phase are all observed during training.

We select the following hashing approaches as the baselines. Iterative Quantization (ITQ) [Gong *et al.*, 2013] and Inductive Hashing (IMH) [Shen *et al.*, 2015b], which are two representative unsupervised hashing approaches. Kernelized Supervised Hashing (KSH) [Liu *et al.*, 2012] and Discrete Supervised Hashing (DSH) [Shen *et al.*, 2015a], which are two celebrated supervised hashing approaches. Deep Hashing Network (DHN) [Zhu *et al.*, 2016] and Deep Neural Network Hashing (DNNH) [Lai *et al.*, 2015], which are two state-of-the-art CNN hashing approaches. Zero-shot Hashing with Transferring Supervised Knowledge (TSK) [Yang *et al.*, 2016] which is one beginning ZSH approach. We implement TSK by ourselves and we use the author provided codes for the others. For all approaches, we use the same training and query sets. For the CNN approaches, we use the same training set to fine-tune their models. For the non-CNN approaches, we adopt the GoogLeNet feature for their input vectors.

We adopt two widely used evaluation metrics in the experiment. The first is mean Average Precision (mAP) based on Hamming ranking. Given a query, all database samples are ranked based on their Hamming distances to the query. The second the Precision within Hamming radius 2 based on look-up table. Given a query, a look-up based retrieval [Shen *et al.*, 2015a] is performed and samples whose Hamming distances to the query are no larger than 2 are returned.

4.3 Training Detail

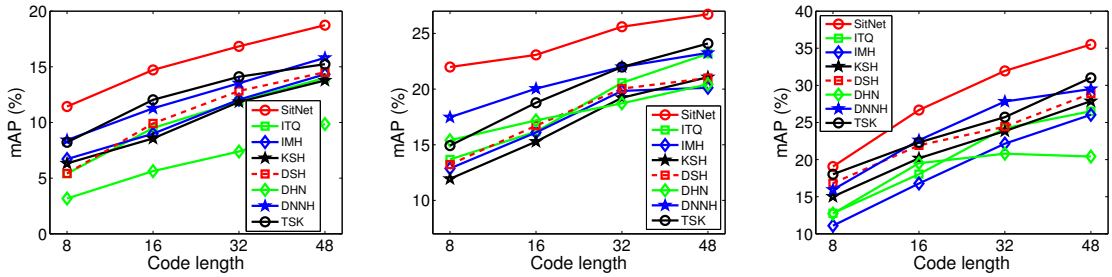
To train our network, we utilize the Caffe [Jia *et al.*, 2014] tool and adopt the AlexNet as the base network by using its convolution and fc layers. In all experiment, the initial learning rate is set to 10^{-3} and the momentum is set to 0.9. The weight decay parameter is 0.0005. The mini-batch size is set to 128. The training terminates at the $50,000$-th iteration. In our model, the max-margin parameter is $\lambda = 1$, the weights of the regularized center loss are $\alpha = \beta = 10^{-3}$, and the weight of the semantic embedding loss is $\gamma = 10^{-2}$.

In addition, since the initial models, including AlexNet and GoogLeNet, are pretrained on ImageNet, they contain knowledge about the 100 categories (both seen and unseen) utilized in our experiment. To better evaluate the zero-shot performance, we retrain the models on the other 900 categories as the base models. For the experiments on ImageNet, all approaches utilize the retrained models instead of the initial ones for further fine-tuning or feature extraction.

4.4 Benchmark Comparison

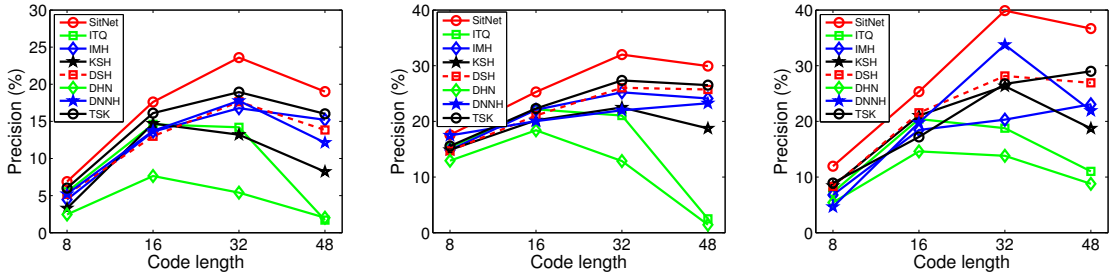
We firstly compare SitNet to the baseline approaches on three benchmarks in the zero-shot scenario. The mAP is plotted in Figure 3 and the precision within Hamming radius 2 is presented in Figure 4. It can be observed that SitNet outperforms the baseline approaches with significant margins on all three datasets, which validates the effectiveness of SitNet for ZSH. Moreover, we also have the following observations.

Firstly, compared to the results in the conventional hashing literatures [Lai *et al.*, 2015; Shen *et al.*, 2015a], the performance of several supervised baseline approaches drops significantly in the zero-shot setting which demonstrates that the existing supervised approaches generalize poorly on unseen concepts. On the other hand, the unsupervised approaches drops less significantly. This is because the unsupervised information, like variance [Gong *et al.*, 2013], is less sensitive to the supervised information. SitNet takes advantage of the



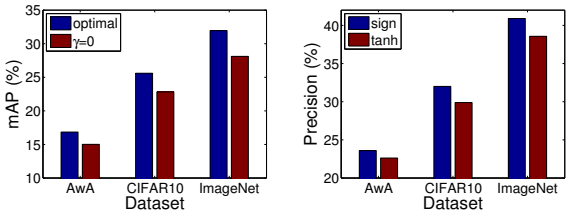
(a) AwA (b) CIFAR10 (c) ImageNet

Figure 3: Mean Average Precision on three datasets.



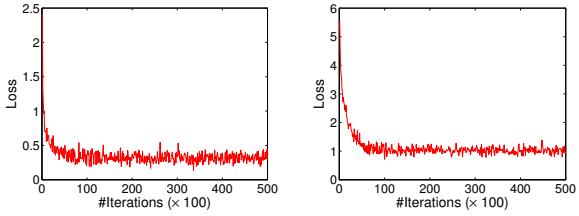
(a) AwA (b) CIFAR10 (c) ImageNet

Figure 4: Precision with Hamming radius 2.



(a) Semantic Embedding (b) Discrete Hashing Layer

Figure 5: Effective verification using 32-bit hashcodes.



(a) CIFAR10 (b) ImageNet

Figure 6: Convergency analysis (32 bits).

semantic vector space for knowledge transfer. In this space the similarity structure among concepts is well captured such that it can transfer the supervised information of seen concepts to the unseen concepts. Even though the model never meets the unseen concepts, it still generalizes well for them.

Secondly, CNN is a powerful tool for many tasks, including the conventional hashing. However, in the zero-shot setting, many CNN hashing approaches only achieves comparable results to the non-CNN ones and some of them even have very unstable performance, like DHN. CNN is capable of discovering the complicated semantic similarity structure if proper supervision is given. However, in the zero-shot setting, it seems that CNN "overfits" the seen concepts such that it may perform poorly for the unseen concepts in some cases. Our SitNet is able to avoid this problem because the semantic vector space helps to improve its generalizability.

4.5 Effectiveness Verification

Effect of Semantic Embedding Loss. The semantic embedding loss is an important difference from the other CNN hashing approaches. In Figure 5(a) we compare the optimal SitNet to the one with $\gamma = 0$ which removes the semantic embedding loss. Obviously, the optimal one outperforms the removed one. The results demonstrate that the semantic embedding loss indeed improves the zero-shot hashing performance.

Effect of Discrete Hashing Layer. Different from previous CNN hashing approaches, SitNet has a discrete hashing layer which directly outputs binary hashcodes. To demonstrate its efficacy, we replace the sign function in this layer with the tanh function. The comparison is presented in Figure 5(b). The results show that the discrete layer performs better than the real-value relaxed model.

In Eq. (6), we propose a simple method to backpropagate

the loss through the discrete hashing layer. In Figure 6, we plot the loss in Eq. (3) w.r.t. the number of training iterations. We can observe that the loss decreases steadily, which demonstrates that the proposed method can well address the discrete optimization problem in the hashing layer.

5 Conclusion

In this paper, we focus on the zero-shot hashing problem and propose a novel SitNet for ZSH which is capable of producing effective hashcodes for both seen and unseen concepts. Specifically, we utilize the semantic vectors of concepts to capture the relationship among concepts and help transfer the supervised knowledge across concepts. A multi-task architecture considering the semantic embedding loss and regularized center loss is adopted. We integrate a discrete hashing layern to prevent information leak and a simple and efficient method is proposed for loss backpropagation. Experiments on three benchmarks demonstrate the efficacy of SitNet.

References

[Akata *et al.*, 2016] Zeynep Akata, Mateusz Malinowski, Mario Fritz, and Bernt Schiele. Multi-cue zero-shot learning with strong supervision. In *CVPR*, 2016.

[Bengio *et al.*, 2013] Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432, 2013.

[Changpinyo *et al.*, 2016] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Synthesized classifiers for zero-shot learning. In *CVPR*, 2016.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[Ding *et al.*, 2016] Guiguang Ding, Yuchen Guo, Jile Zhou, and Yue Gao. Large-scale cross-modality search via collective matrix factorization hashing. *IEEE TIP*, 25(11):5427–5440, 2016.

[Farhadi *et al.*, 2009] Ali Farhadi, Ian Endres, Derek Hoiem, and David A. Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

[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. *IEEE TPAMI*, 2013.

[Guo *et al.*, 2016] Yuchen Guo, Guiguang Ding, Jungong Han, and Xiaoming Jin. Robust iterative quantization for efficient $\ell_{p,q}$-norm similarity search. In *IJCAI*, 2016.

[Guo *et al.*, 2017a] Yuchen Guo, Guiguang Ding, Jungong Han, and Yue Gao. Zero-shot learning with transferred samples. *IEEE TIP*, 26(7):3277–3290, 2017.

[Guo *et al.*, 2017b] Yuchen Guo, Guiguang Ding, Li Liu, Jungong Han, and Ling Shao. Learning to hash with optimized anchor embedding for scalable retrieval. *IEEE TIP*, 26(3):1344–1354, 2017.

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

[Jia *et al.*, 2014] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[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.

[Lampert *et al.*, 2014] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE TPAMI*, 2014.

[Lazaridou *et al.*, 2015] Angeliki Lazaridou, Georgiana Dinu, and Marco Baroni. Hubness and pollution: Delving into cross-space mapping for zero-shot learning. In *ACL*, 2015.

[Lin *et al.*, 2016] Zijia Lin, Guiguang Ding, Jungong Han, and Jianmin Wang. Cross-view retrieval via probability-based semantics-preserving hashing. *IEEE T. Cybernetics*, 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.*, 2016] Haomiao Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. Deep supervised hashing for fast image retrieval. In *CVPR*, 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. *IEEE TIP*, 2015.

[Socher *et al.*, 2013] Richard Socher, Milind Ganjoo, Christopher D. Manning, and Andrew Y. Ng. Zero-shot learning through cross-modal transfer. In *NIPS*, 2013.

[Turian *et al.*, 2010] Joseph P. Turian, Lev-Arie Ratinov, and Yoshua Bengio. Word representations: A simple and general method for semi-supervised learning. In *NIPS*, 2010.

[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*, 104(1):34–57, 2016.

[Wen *et al.*, 2016] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, 2016.

[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.

[Yang *et al.*, 2016] Yang Yang, Yadan Luo, Weilun Chen, Fumin Shen, Jie Shao, and Heng Tao Shen. Zero-shot hashing via transferring supervised knowledge. In *ACM Multimedia*, 2016.

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