



# Deep Supervised Auto-encoder Hashing for Image Retrieval

Sanli Tang, Haoyuan Chi, Jie Yang<sup>(✉)</sup>, Xiaolin Huang,  
and Masoumeh Zareapoor

Institution of Image Processing and Pattern Recognition,  
Shanghai Jiao Tong University, Shanghai, China  
jieyang@sjtu.edu.cn

**Abstract.** Image hashing approaches map high dimensional images to compact binary codes that preserve similarities among images. Although the image label is important information for supervised image hashing methods to generate hashing codes, the retrieval performance will be limited according to the performance of the classifier. Therefore, an effective supervised auto-encoder hashing method (SAEH) is proposed to generate low dimensional binary codes in a point-wise manner through deep convolutional neural network. The auto-encoder structure in SAEH is designed to simultaneously learn image features and generate hashing codes. Moreover, some extra relaxations for generating binary hash codes are added to the objective function. The extensive experiments on several large scale image datasets validate that the auto-encoder structure can indeed increase the performance for supervised hashing and SAEH can achieve the best image retrieval results among other prominent supervised hashing methods.

**Keywords:** Image retrieval · Image hashing · Supervised learning  
Deep neural network · Convolutional auto-encoder

## 1 Introduction

With the growing number of image data on the Internet, fast image retrieval is becoming an increasingly important topic. Image hashing attempts to map higher dimension images to lower dimension binary codes, and thus, the similarity between two sequences can be easily and quickly calculated. Hashing technique, which is the most powerful and important technique in image retrieval, achieves a great success, due to its effectiveness to reduce the cost in term of storage and time.

In previous years, many prominent hashing methods have been proposed [2, 15, 17, 23], including many learning based hashing approaches, see, e.g. [25]. Hashing methods based on handcrafted features (e.g. GIST [20] and HOG [5]) have firstly been studied. Iterative Quantization (ITQ) [13] applies a random

orthogonal transformation to the PCA-projected data and then refines the former orthogonal transformation to minimize quantization error. Kernel-Based Supervised Hashing (KSH) [3] employs a kernel trick to accommodate with the data which are linearly inseparable. [15] proposes to encode the relative order of features rather than quantize the values in ranking subspaces, which can effectively handle prevalent noises in real-world dataset.

In addition, convolution neural networks act as end-to-end methods to extract the features and then to be applied in various tasks. Recently, notable success of deep neural network models [9, 11] in a wide range of areas such as: object detection, image classification, and object recognition, has aroused the researchers' interest to develop hashing methods through deep neural networks. CNNH+ [26] is developed for image hashing and comprises of two efficient stages. In the first stage, it decomposes the similarity matrix into a product of matrix of target hash codes. In the second stage, it builds a convolution network to learn hashing codes from labelled data (if it is on supervised scenario). Later, Deep Supervised Hashing (DSH) [16] prudently combines these two aforementioned stages in CNNH+ into a single network, in which it takes a pair of images with their labels as inputs and attempts to maximize the discriminability of the output space. As a point-wise method which takes single image as the input of network for training, Supervised Semantics-preserving Deep Hashing (SSDH) [27] uses a deep convolution network based on AlexNet [11] to obtain hash codes and directly uses these codes to minimize the classification error. Deep Quantization Network (DQN) [2] proposes a product quantization loss for controlling hashing quality and the quantizability of bottleneck representation.

In the most recent, CNN-based auto-encoder methods [6, 24] emerge as a powerful technique to extract highly abstract features from image data. These extracted features can capture the semantic information of images, which can be used for image hashing in the retrieval task. Several hashing methods based on convolution auto-encoders (CAE) have also been proposed. For example, [21] is one of the most recent method which presents a new hashing method by using variational auto-encoder [7] on unsupervised scenario.

Although many efficient deep supervised hashing methods [2, 15, 16, 27] have been proposed in the last few years, which achieved exciting performance, the studies on supervised auto-encoder structure in image hashing task are limited. En [8] proves the effectiveness of the auto-encoder structure for unsupervised hashing, which encourages the study in this paper on the supervised image hashing by incorporating an auto-encoder structure into a supervised hashing network. The effectiveness of the auto-encoder structure has been proved in image classification [22] and generation [12, 18] tasks, while in this paper, we validate its effectiveness for supervised image hashing method in image retrieval task.

Since the supervised information from image labels is a strong regularization term which drives the images with the same label to be encoded into the same hashing codes, the performance of supervised hashing methods might be limited by the classification accuracy of the supervisory network. However, considering misclassified images by the supervisory network, the auto-encoder structure is

able to restrict those with similar patterns to be encoded with similar hashing codes, which is proved to be effective in unsupervised image hashing task [8, 21] and consequently improves the retrieval results on supervised scenario.

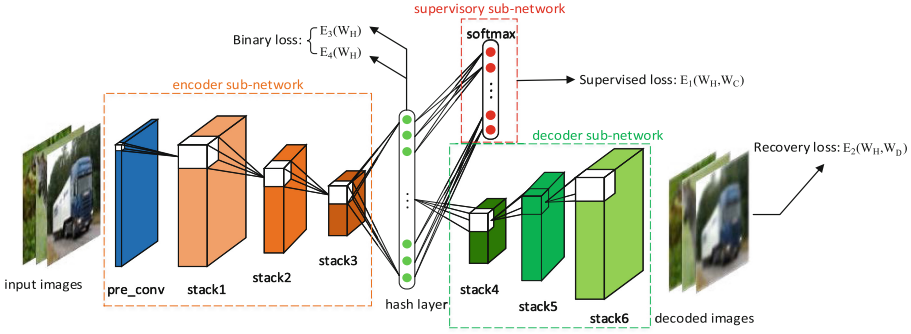
The motivation of this work is straightforward, since the auto-encoder structure has the ability to keep the semantic feature between images. In our work, in order to improve the generalization ability and remedy the overly dependent on the performance of the supervisory network for deep supervised hashing method, we propose a framework based on a supervised auto-encoder hashing (SAEH) model to generate binary hash codes while still keeping their semantic similarities. The auto-encoder structure is also designed to assist the supervisory network to learn more semantic features, which therefore, will increase the semantic information represented by each hashing bit. Following previous works [11, 12], supervised information is incorporated in the deep hashing architecture to associate the hashing bits with the given label, where the mean-square error of original and recovered images and the classification error are simultaneously minimized. In order to convert these codes to binary, some additional relaxations are also incorporated into the objective function. In summary, there are three main contributions of this paper: (1) A framework is proposed to incorporate auto-encoder into supervisory hashing model, which will increase the semantic-keeping and generalization ability. (2) Several typical methods for combining auto-encoder structure with supervised hashing network are inspected to validate their effectiveness in supervised image hashing task. (3) The proposed framework can achieve the best image retrieval results among other prominent supervised hashing methods on several large-scale datasets.

The practicalness and effectiveness of SAEH model are validated through various experiments on MNIST, CIFAR-10, SVHN and UT-Zap50K datasets. In order to statistically compare the performance of the proposed SAEH model and the deep supervised hashing model without the auto-encoder structure, the decoder network with the recovery loss is removed from our hashing model to identify that the effectiveness of auto-encoder. Multiple comparison experiments are also carried out to show the effectiveness of SAEH with other state of art image retrieval methods.

The rest of the paper is organized as follows. Section 2 describes our framework based on the supervised auto-encoder hashing model in detail. Section 3 presents experiments on four large datasets to evaluate the capability of SAEH to generate binary hashing bits. Section 4 gives conclusions of this paper.

## 2 Supervised Auto-encoder Hashing

Let  $\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$  be  $N$  images belonging to labels  $\mathcal{Y} = \{\mathbf{y}_n \in \{0, 1\}^C\}_{n=1}^N$ , where  $C$  is the number of classes. For example, if  $\mathbf{x}_n$  belongs to class  $c_n \in \{1, 2, \dots, C\}$ , we assign its label vector  $\mathbf{y}_n$  as  $(\mathbf{y}_n)_j = 1$  if  $j = c_n$  and 0 otherwise. Then we use  $\mathbf{h}_n \in [0, 1]^K$  to denote the codes generated from  $\mathbf{x}_n$  through an encoder function with the code length  $K$ . Similarly, we denote the binary hashing codes as  $\mathbf{b}_n \in \{0, 1\}^K$  by setting a threshold to  $\mathbf{h}_n$ . In order to obtain hashing



**Fig. 1.** The architecture of SAEH (The source codes of this paper will be public in the future) proposed in this paper, which includes three parts: encoder sub-network, decoder sub-network, and supervisory sub-network. The encoder sub-network is based on ResNet50 [9] where we remove the last two layers and add a fully connected layer to generate hash codes. The  $stackn(n = 1, 2, \dots, 6)$  denotes a group of cascaded residual units as building blocks in [9].

codes from highly abstract features of images, we design a supervised auto-encoder architecture as illustrated in Fig. 1, including: the encoder sub-network, the supervisory sub-network and the decoder sub-network.

### 2.1 Architecture of SAEH

The encoder sub-network is designed to map the normalized input image  $\mathbf{x}_n$  into hashing codes  $\mathbf{h}_n$  in the latent space of SAEH model. We define the parameters in the encoder sub-network as  $W_H$  and the mapping function is signified as  $\mathcal{H} : \mathbf{x}_n \mapsto \mathbf{h}_n$ . Hereafter, in this paper, the output layer of the encoder sub-network is named as hash layer. As the most importance information contained by an image, image labels are used to regularize the latent variables through a supervisory sub-network. The supervisory sub-network takes the latent variables as input, which is generated by the encoder and contains a softmax function:  $\text{softmax}(x)_i = e^{w_i x_i} / \sum_{c=1}^C e^{x_c w_c}$ . It is used to predict the label  $\hat{\mathbf{y}}_n$  of input  $\mathbf{x}_n$  based on its hashing codes  $\mathbf{h}_n$  during training process. And it can be formulated as  $\mathcal{C} : \mathbf{h}_n \mapsto \hat{\mathbf{y}}_n$ , which is parameterized by  $W_C$ . Supervisory sub-network aims to minimize the classification error with the given label  $\mathbf{y}$ , where we calculate the categorical cross-entropy error:

$$l(\mathbf{y}_n, \hat{\mathbf{y}}_n) = - \sum_{c=1}^C y_{nc} \log(\hat{y}_{nc}). \tag{1}$$

And similarly, the loss of the supervisory sub-network among the whole training set can be calculated as follows:

$$\begin{aligned} E_1(W_H, W_C) &= \frac{1}{N} \sum_{n=1}^N l(\mathbf{y}_n, \hat{\mathbf{y}}_n) \\ &= \frac{1}{N} \sum_{n=1}^N l(\mathbf{y}_n, \mathcal{C}(\mathcal{H}(\mathbf{x}_n; W_H); W_C)). \end{aligned} \quad (2)$$

At the same time, the content of an image is important for image hashing, which can be used to improve the generalization ability of the supervised hashing methods and get rid of the heavy dependence on the classification performance. Thus, a decoder sub-network is designed to recover the input image from its hashing codes in the latent space. It can be signified as  $\mathcal{D} : \mathbf{h}_n \mapsto \hat{\mathbf{x}}_n$ , where  $\hat{\mathbf{x}}_n$  is the recovered image and the parameters are represented in term of  $W_D$ . Here, we use mean square error (MSE) between the input images and the decoded images in pixel-wise manner to measure this recovery error as Eq. (3).

$$\begin{aligned} E_2(W_H, W_D) &= \frac{1}{N} \sum_{n=1}^N \|\mathbf{x}_n - \hat{\mathbf{x}}_n\|_2^2 \\ &= \frac{1}{N} \sum_{n=1}^N \|\mathbf{x}_n - \mathcal{D}(\mathcal{H}(\mathbf{x}_n; W_H); W_D)\|_2^2 \end{aligned} \quad (3)$$

## 2.2 Binary Hashing Codes

Considering that the codes generated by encoder  $\mathcal{H}$  are distributed in continuous space, in order to obtain the binary hashing codes, some relaxations should be added to the hash layer. Since we use a sigmoid function:  $\text{sigmoid}(x) = 1/(1 + \exp(-x))$  to activate the output nodes of the hash layer, the output of the hash layer is restricted between 0 and 1.

Following the previous works by [8, 27], we then attempt to convert these activation values into binary values as 0 or 1, which also means to be far from their midpoint. So the relaxation term to get binary hashing codes can be given as following:

$$E_3(W_H) = -\frac{1}{N} \sum_{n=1}^N \|\mathbf{h}_n - 0.5\mathbf{e}\|_2^2, \quad (4)$$

where  $\mathbf{e}$  is a vector with all elements equal to 1.

Moreover, inspired by [27], in order to increase the gap of Hamming distance between the hash codes of the input belonging to different classes, an additional relaxation is added to make sure that hashing codes are as uniformly distributed as possible. Since the latent variables are restricted into  $[0, 1]$  through a sigmoid function, we can regularize the mean of the elements in a sequence of hash codes closer to 0.5 as the mean of the even distribution between  $[0, 1]$ .

$$E_4(W_H) = \sum_{k=1}^K \left( \frac{1}{N} \left( \sum_{n=1}^N h_{nk} \right) - 0.5 \right)^2 \quad (5)$$

The final binary hashing code  $\mathbf{b}_n$  is easily obtained by setting a threshold  $\theta$  ( $\theta = 0.5$  in our work) to  $\mathbf{h}_n$  as Eq. (6), where the quantization error is quite less because the relaxation term in Eq. (4) is incorporated into the objective function for training SAEH model.

$$b_{ni} = \begin{cases} 1 & h_{ni} > \theta \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

### 2.3 Different Ways for Incorporating Autoencoder Structure

Usually, there are three main ways for incorporating an auto-encoder structure into supervised classification tasks: pre-training a supervised classifier network through an auto-encoder structure, simultaneously training classifier with an auto-encoder, and training an auto-encoder model as warm-up for a classifier. The pre-training and warm-up training methods are usually designed to initialize the parameters in the supervised classification networks for fast convergence and improving the performance. Specifically, the pre-training method trains the feature extraction network in the classifier as an encoder with an extra decoder through a few iterations, and then directly removes such decoder network with the recovery error in the objectives and only trains the classifier continuously. Simultaneously training an auto-encoder with a classifier means to train those networks for the whole time during the training process. Training an auto-encoder as warm-up for a classifier attempts to gradually reduce the weight of the decoder with the recovery error during the training process and eventually remove it after some iterations. In this paper, we also investigate the above methods for incorporating auto-encoder structure in supervised hashing model. More formally, those methods for updating the weight  $\gamma$  of the recovery loss term can be summarized as following:

$$\gamma_t = \begin{cases} u(t < t_{\text{pre}}) * \gamma_{\text{init}} & \text{pre-train} \\ \gamma_{\text{init}} & \text{simultaneous train .} \\ \min\{\gamma_{\text{max}}, \max\{\gamma_{t-1} - t \cdot k, 0\}\} & \text{warm-up} \end{cases} \quad (7)$$

In Eq. (7),  $u(\text{condition})$  is an indicator function, which is 1 if the condition is true and 0 otherwise.  $t_{\text{pre}}$  is the iteration times, when the auto-encoder is pre-trained for initialization.  $\gamma_{\text{init}}$  is the initial value of the weight of the recovery loss.  $k$  controls the decreasing speed of the recovery error weight in the warm-up method with  $\gamma_0$  to be  $\gamma_{\text{init}}$ , where the linear decreasing strategy is chosen in our framework and the weight  $\gamma_t$  is clipped into  $[0, \gamma_{\text{max}}]$  after each update.

### 2.4 Objective Function and Implementation

Before going to further discussion, we first formulate the objective function. The final objective function will be obtained by summing all the loss terms,

including the classification loss, the recovery loss, and the relaxations with their corresponding weights and formulated as:

$$\begin{aligned}
 E(W_H, W_D, W_C) &= E_1(W_H, W_C) + \gamma E_2(W_H, W_D) \\
 &\quad + \alpha E_3(W_H) + \beta E_4(W_H) \\
 &\quad + \eta \left( \|W_H\|_2^2 + \|W_D\|_2^2 + \|W_C\|_2^2 \right), \quad (8)
 \end{aligned}$$

where an  $l_2$  regularization term for all of the parameters in SAEH is added during training to reduce the overfitting issue.

SAEH is implemented in Keras [4] with Tensorflow [1] on an NVIDIA GTX 1080 GPU. As shown in Fig. 1, the encoder sub-network is based on ResNet [9], in which we remove the last two layers and add a fully connected layer (hash layer) to generate hashing codes. The supervisory sub-network is connected after the encoder, of which the output layer with  $C$  nodes is directly connected from the hash layer. The decoder network is an inverted architecture of the encoder network, where the bilinear interpolation approach, as the inverted operation of max pooling in the encoder sub-network, is designed to increase the size of the feature maps as up-sampling layer. In addition, we apply the stochastic gradient descent (SGD) method in order to address the problem in Eq. (8) and we set the momentum to 0.9. The learning rate is initialized by 0.1 and reduces 80% every 30 epochs. To validate the effectiveness of auto-encoder structure for supervised hashing, we mainly discuss the influence of recovery weight  $\gamma$  in the following experiments, while other parameters such as  $\alpha$ ,  $\beta$  and  $\eta$  are fixed to be: 0.1, 0.1 and 0.0005 respectively based on some preliminary experiments.

### 3 Experiments

In this section, we carry out various experiments to evaluate the performance of the proposed image hashing framework based on supervised auto-encoder hashing model on several publicly available image datasets. Notice that after the SAEH model being well trained, the supervisory and decoder sub-networks can be simply removed from the framework since only the encoder sub-network is required to generate hashing codes on the test scenario. Thus, our framework is as efficient as other deep supervised hashing models except some extra efforts for training.

#### 3.1 Datasets

- MNIST [14] contains 70k  $28 \times 28$  handwritten images from 0 to 9 in grayscale. Following common splits, we select 6k images per class (60k in total) as training set and the rest as testing set.
- CIFAR-10 [10] consists of 60k  $32 \times 32$  color images of ten common objects. In our experiments, we have divided the images into training set with 5k images per class and the remain for testing set.

- SVHN [19] is a color house number dataset obtained from Google Street View images including 73,257 samples for training and 26,032 for testing.
- UT-Zap50K [28] is a large shoe dataset involving 50,025 images belonging to 4 categories respectively. We randomly select 46,025 samples for training and the rest 4,000 samples for testing.

**Table 1.** mAP@1000(%) and precision rate (with Hamming radius to be 2) on CIFAR-10 dataset w.r.t different  $\gamma$  ( $\alpha = \beta = 0.1$ )

$\gamma$	0	0.01	0.1	1.0	10
mAP@1000	84.56	87.30	<b>87.35</b>	87.00	86.13
Precision	88.16	88.77	88.90	91.93	<b>95.59</b>

**Table 2.** mAP@1000(%), precision rate(%) (with Hamming radius to be 2), and the classification accuracy (as a reference) on CIFAR-10 and UT-Zap50K datasets with 32 bits w.r.t different methods to incorporate auto-encoder structure into supervised hashing model. Notice that the evaluation result by accuracy is the classification accuracy of the supervisory sub-network.

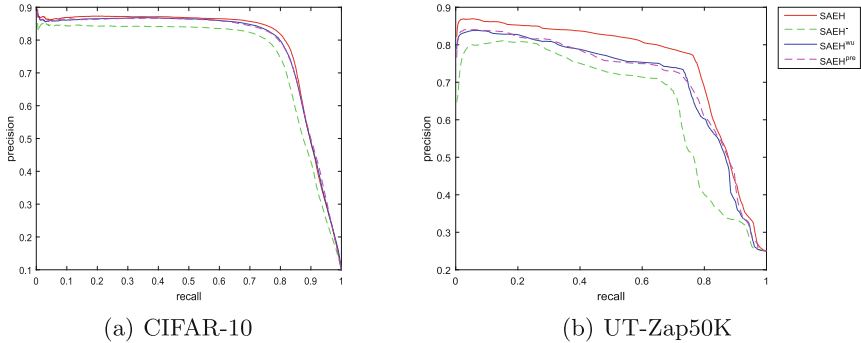
Method	CIFAR-10			UT-Zap50K		
	mAP@1000	Precision	Accuracy	mAP@1000	Precision	Accuracy
SAEH	<b>87.35</b>	88.90	0.8868	<b>85.74</b>	<b>88.35</b>	<b>0.8135</b>
SAEH <sup>-</sup>	84.56	88.16	<b>0.9110</b>	80.70	75.59	0.8080
SAEH <sup>pre</sup>	86.52	<b>89.3</b>	0.8873	83.78	79.94	0.7970
SAEH <sup>wu</sup>	86.59	88.61	0.8938	83.65	79.56	0.7959

### 3.2 Ablation Study for Auto-encoder Structure

To evaluate the effectiveness of the decoder structure in SAEH model, where we simultaneously train an auto-encoder with a classifier, for alleviating the dependence on the classification accuracy, we apply SAEH and contrastive method (denoted as SAEH<sup>-</sup>, where we remove the decoder structure from SAEH by setting the weight of recovery error  $\gamma$  to 0 in the Eq. (8)) on the MNIST, CIFAR-10 and SVHN datasets respectively. The retrieval results of SAEH and SAEH<sup>-</sup> measured by mAP@1000 are shown in the last two rows in Table 3. Comparing to SAEH<sup>-</sup> model, where the decoder sub-network with the recovery loss is ignored during training, SAEH increases the mAP around 0.6%–3.64% on different datasets with the help of decoder structure. Moreover, the weight on the decoder loss  $\gamma$  is inspected to evaluate influences of the decoder network on the performance in supervised image retrieval tasks. The retrieval results on CIFAR-10 dataset of 32 bits with various  $\gamma$  are illustrated in Table 1, where



SAEH achieves best retrieval result under the measurement of mAP@1000 when  $\gamma = 0.1$  and the precision rate with Hamming distance lower than 2 when  $\gamma = 10.0$ . Besides, the model with  $\gamma = 0$  by removing the auto-encoder structure has the worst behaviour among the models with weight  $\gamma > 0$ , which also validates the advantage of the auto-encoder structure in supervised image hashing task. As a trade off, we assign  $\gamma$  to 1 in the following experiments.



**Fig. 2.** Precision-recall (PR) curves of the four methods for incorporating auto-encoder structure into supervised hashing model on (a) CIFAR-10 and (b) UT-Zap50K datasets, with hashing bit length to be 32.

### 3.3 Ablation Study for Incorporating Auto-encoder Methods

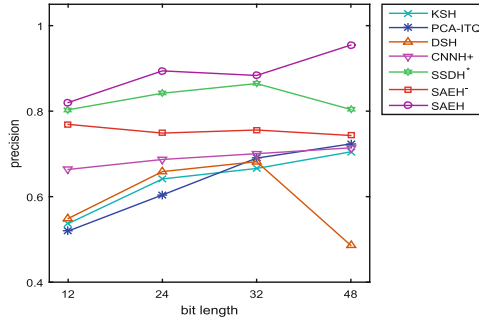
We carry out some experiments to evaluate the effectiveness of the three different methods in Eq. (7) for incorporating auto-encoder structure in supervised hashing model. We initialize the hyperparameters  $\gamma_{\text{init}}$ ,  $t_{\text{pre}}$  and  $k$  as 0.1, 2k, 0.0001 respectively while other parameters in the objective function remains the same. We denote the three methods in Eq. (7): pre-training, simultaneously training and warm-up training as SAEH<sup>pre</sup>, SAEH (corresponding with the notation in other experiments), and SAEH<sup>wu</sup> respectively. We compare those methods on the CIFAR-10 and UP-Zap50K datasets with bit length to be 32. The experiment results measured by mAP@1000 and precision rate with Hamming radius to be 2 are in Table 2. For comparison, we also add the result without the auto-encoder structure as SAEH<sup>-</sup> in the table. The classification accuracy by the supervisory sub-network is also appended as the reference.

From Table 2, we can find that the auto-encoder structure indeed increases the effective of supervised hashing methods. Although the SAEH<sup>-</sup> with out the auto-encoder structure achieves the excellent results under the evaluation of the classification accuracy, comparing to the other three variants with auto-encoder structure, it has the worst behaviour for supervised image hashing under the measurement of both the mAP@1000 and precision rate with Hamming radius to be 2, which verifies that the auto-encoder structure can alleviate the overly dependant on the classification accuracy in supervised hashing methods.

The experiments results in Table 2 also shows the advantage of simultaneously training auto-encoder with a classifier, which achieves the best results on both CIFAR-10 and UT-Zap50K datasets according to criterion of mAP@1000. In other words, when the recall rate decreases, the precision rate of the simultaneously training method increases faster than other incorporating methods as well as the supervised-only SAEH<sup>-</sup>, which is illustrated in Fig. 2 as the precision recall curves on CIFAR-10 and UT-Zap50K datasets, respectively.

**Table 3.** mAP@1000(%) of SAEH, supervised-only hashing (denoted as SAEH<sup>-</sup>) and other advanced hashing methods w.r.t. different number of hashing bits on MNIST, CIFAR-10 and SVHN datasets.

Method	MNIST				CIFAR-10				SVHN			
	12	24	32	48	12	24	32	48	12	24	32	48
KSH [3]	24.30	36.63	31.10	33.25	17.65	14.80	15.50	16.63	24.18	24.36	24.72	21.87
ITQ [13]	37.63	53.87	51.76	54.11	12.93	14.06	13.40	15.11	16.22	16.85	19.67	19.89
DSH [16]	96.05	97.35	98.10	98.13	38.17	38.70	40.19	37.29	73.16	70.33	82.16	77.33
CNNH+ [26]	97.57	97.89	98.04	98.33	40.00	42.00	44.89	44.55	78.32	81.46	81.81	84.00
DQN [2]	98.02	98.16	98.22	98.06	55.40	55.80	56.40	58.00	86.77	86.80	87.01	86.89
SSDH* [27]	98.83	98.97	98.96	99.15	82.31	84.07	83.78	84.28	93.19	93.98	93.95	94.46
SAEH <sup>-</sup>	99.30	99.38	99.38	99.41	84.30	85.71	84.56	82.69	94.91	95.55	95.86	96.14
SAEH	99.50	99.44	99.53	99.54	85.85	87.38	87.00	86.33	95.48	96.28	95.95	96.45



**Fig. 3.** Precision rate (with Hamming distance  $r = 2$ ) of different hashing methods on UT-Zap50K dataset w.r.t different length of hash bits.

### 3.4 Evaluation on SAEH and Other Methods

In this experiment, we compare our proposed framework based on a supervised auto-encoder hashing model with other prominent hashing methods to verify the effectiveness and competitiveness of our framework in supervised image hashing task. We adopt the simultaneously training auto-encoder with a supervisory sub-network mentioned above as our SAEH model. For KSH and ITQ methods based on handcraft features, we first calculate the 512 GIST features of each image for

training. In the second stage of CNNH+ [26], we follow the authors' scheme to carefully design a convolution network for generating hashing codes based on image label information. So far as we know, SSDH [27] is the most prominent supervised hashing method. For fair comparison, we also re-implement SSDH by using ResNet50 [9] as its inference model, denoted as SSDH\*. For convincing evaluation, we repeat the experiments for each hashing method mentioned above for 5 times and illustrate the average results. The retrieval results of mAP@1000 on MNIST, CIFAR-10 and SVHN datasets are shown in Table 3 and the precision rate with Hamming distance  $r=2$  on UT-Zap50K dataset is shown in Fig. 3. From Table 3, comparing to the results from other competitive hashing methods, SAEH increases the mAP@1000 from 0.39% to 3.54%, while it also increases the precision rate with Hamming radius to be 2 from 1.7% to 15.1% on UT-Zap50K dataset in Fig. 3, which validates of the effectiveness and practicalness of the proposed supervised hashing framework based on the auto-encoder structure.

## 4 Conclusion

The performance of supervised hashing methods are always limited by the classification accuracy of the classifier in the model. Consider that the semantic information in the image can be captured efficiently through an auto-encoder structure, which is able to improve the performance of the supervised hashing methods by alleviating the dependence on the accuracy of classification sub-network. Therefore, in this paper, we propose a hashing framework based on a supervised auto-encoder model, which learns semantic preserving hashing codes of images. Moreover, some extra relaxations are introduced that turn the output of hash layer into binary codes and increase the gap of the Hamming distance between classes. Experiment results prove that SAEH takes both the advantage of supervisory and auto-encoder networks and performs better than the contrastive model without the decoder structure. The equilibrium of the recovery loss and supervisory loss is also inspected in this paper. The extended experiments on three main methods to incorporate auto-encoder structure into supervised hashing show the superior effectiveness of simultaneously training an auto-encoder and a supervisory sub-network. Moreover, comparing to other state-of-art methods in supervised image retrieval task, the proposed framework SAEH achieves superior retrieval performance and provides a promising architecture for deep supervised image hashing.

## References

1. Abadi, M., Agarwal, A., Barham, P., et al.: TensorFlow: large-scale machine learning on heterogeneous distributed systems (2016)
2. Cao, Y., Long, M., Wang, J., Zhu, H., Wen, Q.: Deep quantization network for efficient image retrieval. In: Thirtieth AAAI Conference on Artificial Intelligence, pp. 3457–3463 (2016)

3. Chang, S.F., Jiang, Y.G., Ji, R., Wang, J., Liu, W.: Supervised hashing with kernels. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2074–2081 (2012)
4. Chollet, F.: Keras (2015). <https://github.com/fchollet/keras>
5. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2005, pp. 886–893 (2005)
6. Dilokthanakul, N., et al.: Deep unsupervised clustering with gaussian mixture variational autoencoders (2016)
7. Doersch, C.: Tutorial on variational autoencoders. arXiv preprint [arXiv:1606.05908](https://arxiv.org/abs/1606.05908) (2016)
8. En, S., Crémilleux, B., Jurie, F.: Unsupervised deep hashing with stacked convolutional autoencoders. Working paper or preprint, May 2017. <https://hal.archives-ouvertes.fr/hal-01528097>
9. He, K., Zhang, X., Ren, S.: Deep residual learning for image recognition. In: Computer Vision and Pattern Recognition, pp. 770–778 (2016)
10. Krizhevsky, A.: Learning multiple layers of features from tiny images (2009)
11. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: International Conference on Neural Information Processing Systems, pp. 1097–1105 (2012)
12. Larsen, A.B.L., Snderby, S.K.: Larochelle: autoencoding beyond pixels using a learned similarity metric, pp. 1558–1566 (2015)
13. Lazebnik, S.: Iterative quantization: a procrustean approach to learning binary codes. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 817–824 (2011)
14. Lcun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proc. IEEE* **86**(11), 2278–2324 (1998)
15. Li, K., Qi, G.J., Ye, J., Yusuph, T., Hua, K.A.: Semantic image retrieval with feature space rankings. *Int. J. Semant. Comput.* **11**(2), 171–192 (2017)
16. Liu, H., Wang, R., Shan, S., Chen, X.: Deep supervised hashing for fast image retrieval. In: Computer Vision and Pattern Recognition, pp. 2064–2072 (2016)
17. Liu, W., Wang, J., Kumar, S., Chang, S.F.: Hashing with graphs. In: International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, 28 June–July, pp. 1–8 (2011)
18. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I.: Adversarial autoencoders. *Computer Science* (2015)
19. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: Nips Workshop on Deep Learning and Unsupervised Feature Learning (2011)
20. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. *Int. J. Comput. Vis.* **42**(3), 145–175 (2001)
21. Pu, Y., Gan, Z., Heno, R.: Variational autoencoder for deep learning of images, labels and captions (2016)
22. Rasmus, A., Berglund, M., Honkala, M., Valpola, H., Raiko, T.: Semi-supervised learning with ladder networks. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) *Advances in Neural Information Processing Systems* 28, pp. 3546–3554. Curran Associates, Inc. (2015). <http://papers.nips.cc/paper/5947-semi-supervised-learning-with-ladder-networks.pdf>
23. Shao, J., Wu, F., Ouyang, C., Zhang, X.: Sparse spectral hashing. *Pattern Recognit. Lett.* **33**(3), 271–277 (2012)

24. Vincent, P., Larochelle, H., Lajoie, I., et al.: Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.* **11**(12), 3371–3408 (2010)
25. Wang, J., Zhang, T.: A survey on learning to hash. *IEEE Trans. Pattern Anal. Mach. Intell.* **PP**(99), 1 (2016)
26. Xia, R., Pan, Y., Lai, H., Liu, C.: Supervised hashing for image retrieval via image representation learning. In: *AAAI*, vol. 1, p. 2 (2014)
27. Yang, H.F., Lin, K., Chen, C.S.: Supervised learning of semantics-preserving hash via deep convolutional neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **PP**(99), 1 (2015)
28. Yu, A., Grauman, K.: Fine-grained visual comparisons with local learning. In: *Computer Vision and Pattern Recognition*, pp. 192–199 (2014)