Discriminative Deep Hashing for Scalable Face Image Retrieval

Jie Lin, Zechao Li, Jinhui Tang[*]
School of Computer Science and Engineering, Nanjing University of Science and Technology
jinhuitang@njust.edu.cn

Abstract

With the explosive growth of images containing faces, scalable face image retrieval has attracted increasing attention. Due to the amazing effectiveness, deep hashing has become a popular hashing method recently. In this work, we propose a new Discriminative Deep Hashing (DDH) network to learn discriminative and compact hash codes for large-scale face image retrieval. The proposed network incorporates the end-to-end learning, the divide-and-encode module and the desired discrete code learning into a unified framework. Specifically, a network with a stack of convolution-pooling layers is proposed to extract multi-scale and robust features by merging the outputs of the third max pooling layer and the fourth convolutional layer. To reduce the redundancy among hash codes and the network parameters simultaneously, a divide-and-encode module to generate compact hash codes. Moreover, a loss function is introduced to minimize the prediction errors of the learned hash codes, which can lead to discriminative hash codes. Extensive experiments on two datasets demonstrate that the proposed method achieves superior performance compared with some state-of-the-art hashing methods.

1 Introduction

With the growing popularity of social networking on intelligent mobile services, the number of images containing faces has witnessed an explosive increase in recent years. Consequently, face image retrieval, which aims to identify images containing the person in the given face image, has become an attractive research area. The main challenges of face image retrieval are large intra-class variations and the cost of computing time and storage. Therefore, it is necessary to develop effective face image retrieval methods to address the above two problems.

To tackle the first problem, previous methods mainly focus on how to find better hand-crafted visual descriptors, such as LBP [Ojala *et al.*, 2002], to represent visual contents of face images. Recently, deep learning, such as Convolutional Neural Network (CNN), has shown its amazing performance in many computer vision tasks, such as image classification [Krizhevsky *et al.*, 2012; Li *et al.*, 2015] and face recognition [Sun *et al.*, 2014a]. CNN features learned from images are more robust and well capture the potential semantic structure of images. And many methods have been proposed to use the CNN features for vision tasks [Sun *et al.*, 2014b]. However, for large-scale face image retrieval, methods based on CNN features are still high dimensional and inefficient for the retrieval task. As a consequence, hashing methods, especially deep hashing methods, have been widely studied to map high-dimensional face representations to compact binary codes. Due to the powerful ability of deep features and retrieval effectiveness, deep hashing methods have attracted much attention recently.

Deep hashing methods have been proposed for image retrieval [Xia *et al.*, 2014; Liu *et al.*, 2016; Lai *et al.*, 2015; Dong *et al.*, 2016; Tang *et al.*, 2017]. These methods mainly take the supervised information encoded in pairs or triplets of images as training inputs. Some deep hashing methods learn hash codes and hash functions separately, such as CNNH [Xia *et al.*, 2014]. CNNH first learns hash codes by preserving the supervised pairwise similarity, and then learns hash functions based on the learned hash codes using a deep convolutional network. To incorporate the hash learning and the deep network into an end-to-end system, some other deep hashing methods have been proposed [Liu *et al.*, 2016; Lai *et al.*, 2015]. DSH [Liu *et al.*, 2016] learns hash codes based on a CNN architecture by preserving the similarity encoded in the input pairs of images (similar/dissimilar). NINH [Lai *et al.*, 2015] learns hash codes by minimizing the triplet ranking loss based on the "network in network" network [Lin *et al.*, 2013]. In [Dong *et al.*, 2016], the low-rank hashing first pre-learns hash functions based on the CNN features and introduces the fine-tuning procedure based on the triplet ranking loss. These methods mainly preserve the pairwise similarity or the triplet rank. It is expensive and time-consuming to obtain these pairs or triplets. And the optimization complexity of models would greatly increase and the learned hash codes cannot guarantee to be discriminative. Moreover, the redundancy among hash codes is not explored.

Towards this end, we propose a new Discriminative Deep Hashing (DDH) method to learn discriminative and compact

[*]Corresponding author

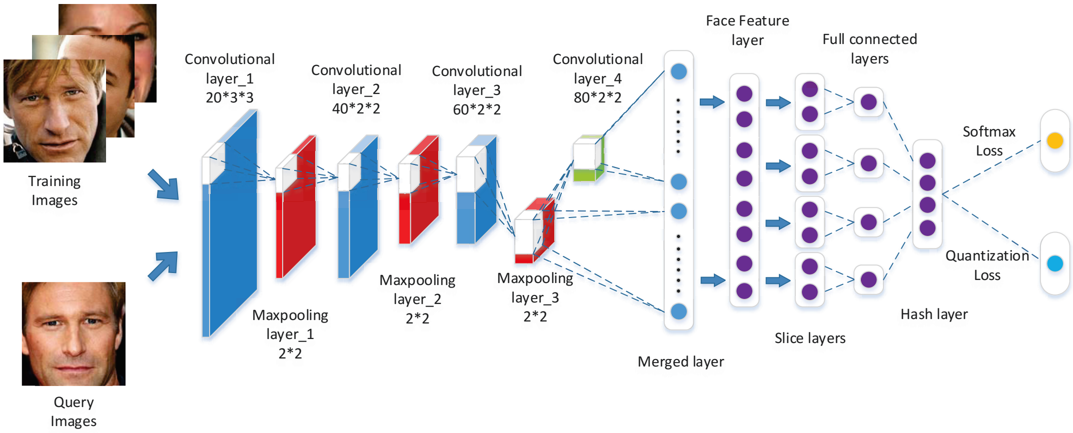


Figure 1: Illustration of the proposed DDH framework for large-scale face image retrieval.

hash codes for scalable face image retrieval, as shown in Figure 1. The proposed method is an end-to-end system by incorporating the feature extraction and hash learning into a unified framework. Specifically, a sub-network with a stack of convolution-pooling layers is built to take face images associated with labels as inputs and extract features for each face image. To obtain multi-scale and robust facial features, the outputs of the third max pooling layer and the fourth convolutional layer are merged. Second, a divide-and-encode module is introduced to reduce the redundancy among hash codes and network parameters simultaneously. To make the learned hash codes discriminative, the discrete hash codes are required to directly predict the labels of face images. To well solve the proposed optimization formulation, the discrete hash codes are relaxed to be continuous and a regularizer is introduced to approximate the desired discrete values. To validate the effectiveness of the proposed DDH method, extensive experiments are conducted on two face image datasets. Experimental results compared with some state-of-the-art hashing methods show the promising performance of the proposed method.

The remainder of the paper is organized as follows. In Section 2, we review the related works about deep hashing methods and face-related tasks via deep learning network. Section 3 elaborates the proposed DDH method for large-scale face image retrieval. Then, we conduct experiments to show the effectiveness of the proposed deep hashing method in Section 4. Finally, Section 5 concludes this work.

2 Related Work

In this section, we overview previous related works about deep hashing methods and face-related tasks via deep learning network.

2.1 Deep Hashing Methods

Existing learning-based hashing methods can be divided into three categories: supervised hashing [Xia *et al.*, 2014; Shen *et al.*, 2015; Liu *et al.*, 2016; Tang *et al.*, 2017], semi-supervised hashing, unsupervised hashing [Andoni and Indyk, 2006; Weiss *et al.*, 2008; Tang *et al.*, 2015]. Most existing deep hashing methods belong to supervised hashing methods. In the pipe-lines of these methods, they project the high-dimensional feature space to the low-dimensional Hamming space by a deep network. And additional appropriate constraints such as point-wised constraints [Lin *et al.*, 2015], pair-wised constraints [Lai *et al.*, 2015; Liu *et al.*, 2016; Zhu *et al.*, 2016], ranking-wised constraints [Zhao *et al.*, 2015], are imposed on the networks to generate discriminative binary hash codes.

Some deep hashing methods learn hash codes and hash functions separately, such as CNNH [Xia *et al.*, 2014]. They do not belong to the end-to-end methods. CNNH uses two stages to learn hash codes and hash functions. It first learns hash codes by preserving the supervised pairwise similarity, and then learns hash functions based on the learned hash codes using a deep convolutional network. To build an end-to-end deep learning system, some other deep hashing methods have been proposed [Zhao *et al.*, 2015; Liu *et al.*, 2016; Lai *et al.*, 2015]. In [Zhao *et al.*, 2015], hash functions are learned for multi-label image retrieval by exploring the multilevel semantic ranking supervised information under a deep CNN architecture. In [Lai *et al.*, 2015], NINH is proposed to improve CNNH by fusing the hash learning into the deep structure. NINH learns hash codes by minimizing the triplet ranking loss based on the “network in network” network [Lin *et al.*, 2013]. DSH [Liu *et al.*, 2016] learns hash codes by preserving the similarity encoded in the input pairs of images (similar/dissimilar) based on a CNN architecture. DHN [Zhu *et al.*, 2016] learns image representation tailored to hash coding by using a pairwise crossentropy loss layer for similarity-preserving learning. In [Dong *et al.*, 2016], the low-rank hashing first pre-learns hash functions based on the CNN features and introduces the fine-tuning procedure based on the triplet ranking loss. These methods mainly take the supervised pairs or triplets as training inputs. However, It is expensive and time-consuming to obtain these pairs or triplets. And the optimization complexity of models would greatly increase and the learned hash codes cannot guarantee to be discriminative. Moreover, the redundancy among hash codes is not explored.

Different from the above deep hashing methods, we propose a novel deep hashing method by taking face images associated with labels as training inputs. Feature extraction and hash learning are integrated into a unified deep architecture. The learned hash codes are expected to well predict the labels of face images. To reduce the redundancy among hash codes and network parameters simultaneously, the proposed deep framework leverages a divide-and-encode module.

2.2 Face-related Tasks via Deep Learning Network

Recently, inspired from the advancement of deep learning, numerous face-related tasks choose to employ deep learning networks and achieve significant results. In [Taigman *et al.*, 2014], the 3D model is utilized to align faces before training, and several locally connected layers without weight sharing are used to improve the precision instead of the standard convolutional layers. DeepID is proposed to extract multi-scale facial features from different face regions through a deep network, and indeed achieves terrific face verification accuracy on the LFW dataset [Sun *et al.*, 2014a]. In [Schroff *et al.*, 2015], the triplet loss instead of the softmax loss is introduced to minimize the distance between an anchor and a positive data point, and maximize the distance between an anchor and a negative data point. A new loss function, named center loss, is proposed to minimize the intra-class distances based on the deep features [Wen *et al.*, 2016]. The center loss is more convenient and can save more space and time than the contrastive loss and triplet loss. Similar to [Sun *et al.*, 2014b], we utilize the deep network architecture as shown in Figure 1 to extract robust and multi-scale facial features.

3 The Proposed DDH Approach

In this section, we will introduce the proposed DDH method in details. The motivation of this work is first introduced and then the proposed model is elaborated.

3.1 Motivation

The essential problem of hashing is how to learn discriminative and compact hash codes, which can significantly improve the performance of image retrieval. Due to the powerful ability of CNN for learning data representations, we focus on proposing a deep hashing method by guaranteeing that the hash learning is optimally compatible with the feature representation learning, as shown in Figure 1.

To extract facial features hierarchically, a deep network is developed based on the CNN architecture with convolutional layers and max-pooling layers. The first three full-connected convolutional layers mainly extract mid-level representations from the original pixels, such as simple edges and textures [Taigman *et al.*, 2014]. The outputs of convolution networks become more robust to local translations by introducing max-pooling layers. More global and high-level features are gradually formed in the top layers, and the local stability assumption of the convolution does not exist. Thus, the fourth layer is instead locally connected [Sun *et al.*, 2014b] and the weights of the fourth convolutional layer are totally unshared. To extract robust and multi-scale facial representations, the outputs of the third max-pooling layer and the fourth convolutional layer are merged into the Merged Layer, and a full-connected convolutional layer, i.e., the Face Feature layer, is introduced as shown in Figure 1.

The ideal hash codes should be directly learned by the deep network. Thus, the outputs of the proposed deep network are the desired discrete hash codes. To make the hash codes discriminative, the learned hash codes are required to enable to well predict the labels of face images. On the other hand, the learned hash codes should be compact. To reduce the redundancy among hash codes, the proposed deep network leverages a divide-and-encode module. The divide-and-encode module can also decrease the number of the network parameters, which can reduce the computing cost.

3.2 The Proposed Formulation

The goal of hash learning is to learn a hash function $h : \mathbb{R} \mapsto \{-1, 1\}$ that map images into the Hamming space.

Suppose we have N training face images $\{\mathbf{x}_i\}_{i=1}^N$ associated with its label vector $\mathbf{y}_i \in \mathbb{R}^M$, where M denotes the number of class labels. And $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_N] \in \mathbb{R}^{M\times N}$ denotes the label matrix. If $\mathbf{x}_i$ belongs to the j-th class, $Y_{ji} = 1$ and 0 otherwise. The goal of the proposed DDH method is to learn discrete hash codes $\mathbf{b}_i \in \{-1, 1\}^K$ for each face image $\mathbf{x}_i$ where K denotes the length of the hash codes. And the learned hash codes $\mathbf{b}_i$ are discriminative to well predict its label. To this end, we propose to learn hash codes by an end-to-end deep network as shown in Figure 1.

In the developed deep network, a sub-network with a stack of convolution-pooling layers extracts multi-scale and robust facial features. The first three full-connected convolutional layers associated with max-pooling layers mainly extract mid-level representations from the original pixels, such as simple edges and textures. The fourth convolutional layer can obtain more global and high-level features. To obtain robust and multi-scale facial representations, the outputs of the third max-pooling layer and the fourth convolutional layer are merged into the Merged Layer. And then a Face Feature layer is introduced by fully connecting the outputs of the Merged Layer. To reduce the number of network parameters and the redundancy among the desired hash codes simultaneously, a divide-and-encode module is introduced similar to [Lai *et al.*, 2015]. It equally splits the intermediate features obtained from the Face Feature layer into K groups. Then we map each group to one real value by a full-connected layer. Finally, we merge all K real values into a Hash layer. The outputs of the Hash layer is the desired discrete hash codes $\mathbf{B} \in \{-1, +1\}^{K\times N}$.

$$\mathbf{B} = g(\mathbf{X}) \in \{-1, +1\}^{K\times N} \tag{1}$$

Here $\mathbf{X}$ is the data matrix containing N face images and $g(\cdot)$ denotes the transformations of the deep network. The proposed deep hashing network adopts an end-to-end learning framework to jointly learn good representations and compact binary hash codes.

To enhance the discriminative ability of the learned hash codes in the supervised case, it is necessary to introduce loss functions to measure the predictive power of the learned hash codes. Some methods use the contrastive loss [Sun *et al.*, 2014a] and triplet ranking loss [Schroff *et al.*, 2015]. For the contrastive loss, sufficient pairs of images should be constructed, which easily leads to the imbalanced problem of positive and negative pairs. The triplet ranking loss needs complex sampling strategies to obtain suitable triplets and results in a more difficult optimization problem. They can also result in bigger computing cost including time and space. Different from them, we take the face images with their class labels as training inputs. To guarantee the discriminative power of the hash codes, the learned hash codes are expected to well predict the corresponding class labels. For simplicity, we employ the softmax function as the prediction function, which is the generalization of the logistic function.

$$p(Y_{ij}=1|\mathbf{x}_i)=\frac{e^{\mathbf{w}_j^T\mathbf{b}_i}}{\sum_{k=1}^{M}e^{\mathbf{w}_k^T\mathbf{b}_i}} \tag{2}$$

Here $\mathbf{w}_k$ is the linear prediction function for the k-th class. The loss function over all the training images is as follows.

$$\begin{aligned}\ell_S&=\sum_{i=1}^{N}\sum_{j=1}^{M}-Y_{ij}\log p(Y_{ij}=1|\mathbf{x}_i)+\frac{\alpha}{2}\|\mathbf{W}\|_F^2\\&=\sum_{i=1}^{N}\sum_{j=1}^{M}-Y_{ij}\log\frac{e^{\mathbf{w}_j^T\mathbf{b}_i}}{\sum_{k=1}^{M}e^{\mathbf{w}_k^T\mathbf{b}_i}}+\frac{\alpha}{2}\|\mathbf{W}\|_F^2\end{aligned} \tag{3}$$

To avoid overfitting and enhance the generalization ability, we introduce a regularization term into the above loss function. $\mathbf{W}=[\mathbf{w}_1,\cdots,\mathbf{w}_M]$, α is a nonnegative hyperparameter, and $\|\cdot\|_2$ is the Frobenius norm of a matrix. Consequently, the proposed formulation is obtained.

$$\begin{aligned}&\min\sum_{i=1}^{N}\sum_{j=1}^{M}-Y_{ij}\log\frac{e^{\mathbf{w}_j^T\mathbf{b}_i}}{\sum_{k=1}^{M}e^{\mathbf{w}_k^T\mathbf{b}_i}}+\frac{\alpha}{2}(\|\mathbf{W}\|_F^2+\Omega)\\&\text{s.t.}\quad \mathbf{b}_i\in\{-1,+1\}^K\end{aligned} \tag{4}$$

To enhance the generalization ability of the network, the regularizer Ω of all the parameters is also introduced.

If we can directly solve the optimization problem (Eqn. 4), it would be perfect to obtain the discrete hash codes and hash functions. Unfortunately, the above problem (Eqn. 4) is difficult to be optimized due to the discrete constraints. Inspired from [Liu *et al.*, 2016], we relax $\mathbf{B}$ to be continuous and utilize the *tanh* function to to limit the output values within the range (-1,1). Besides, we impose a regularizer on the continuous values to approach the desired discrete values (+1/-1). Then, the problem (Eqn. 4) is rewritten as follows.

$$\begin{aligned}&\min\sum_{i=1}^{N}\sum_{j=1}^{M}-Y_{ij}\log\frac{e^{\mathbf{w}_j^T\mathbf{b}_i}}{\sum_{k=1}^{M}e^{\mathbf{w}_k^T\mathbf{b}_i}}+\frac{\alpha}{2}(\|\mathbf{W}\|_F^2+\Omega)\\&+\beta\sum_{i=1}^{N}\||\mathbf{b}_i|-\mathbf{1}\|_1\end{aligned} \tag{5}$$

Here $|\cdot|$ is the element-wise absolute value operation, $\mathbf{1}$ is a vector of ones and $||\cdot||_1$ is the ℓ_1-norm of one vector. β is a weighting parameter to control the importance of the regularizer. The above optimization problem (Eqn. 5) enables to learn discriminative and compact hash codes by leveraging the end-to-end learning, divide-and-encode module, hashing learning and prediction function learning simultaneously.

To optimize the proposed formulation, the developed network is trained by using back-propagation scheme with minibatch gradient descent algorithm. In practice, the proposed model is implemented based on the Keras library with the Theano backend [Bergstra *et al.*, 2010]. Under this platform, what we need to do is just to input the objective function, and Theano will complete the training with its characteristic of automatic derivation.

Once the developed deep hashing network is trained, we can use it to generate a K-bit hash codes for any input image. Based on the outputs of the Hash layer, the hash codes can be calculated by using the $\mathrm{sgn}(\cdot)$ function: $\mathrm{sgn}(b)=1$ if $b>0$ and $\mathrm{sgn}(b)=-1$ otherwise.

4 Experiments

To evaluate the effectiveness of the proposed deep hashing method for scalable face image retrieval, we conduct experiments on two widely used large-scale face image dataset: YouTube Faces [Wolf *et al.*, 2011] and FaceScrub [Ng and Winkler, 2014], and compare the proposed DDH method with several state-of-art hashing methods.

4.1 Datasets and Evaluation Metrics

Experiments are condcuted on two face image dataset: YouTube Faces [Wolf *et al.*, 2011] and FaceScrub [Ng and Winkler, 2014].

YouTube Faces. This dataset contains 3,425 videos involved with 1,595 different persons. The shortest clip duration is 48 frames, the longest clip is $6,070$ frames, and the average length of a video clip is 181.3 frames. We randomly selected 40 face images per person as the training set and 5 images per person as the test set. Thus, we have 63800 images as the training set and 7975 images as the test data. All face images are resized to 32 × 32.

FaceScrub. It composes a total of 106,863 face images of 530 celebrities, with about 200 images per person. So far, it is one of the largest public face datasets. We randomly select 5 face images per person as the test data, and the remainder as the training set. All face images are also resized to 32×32.

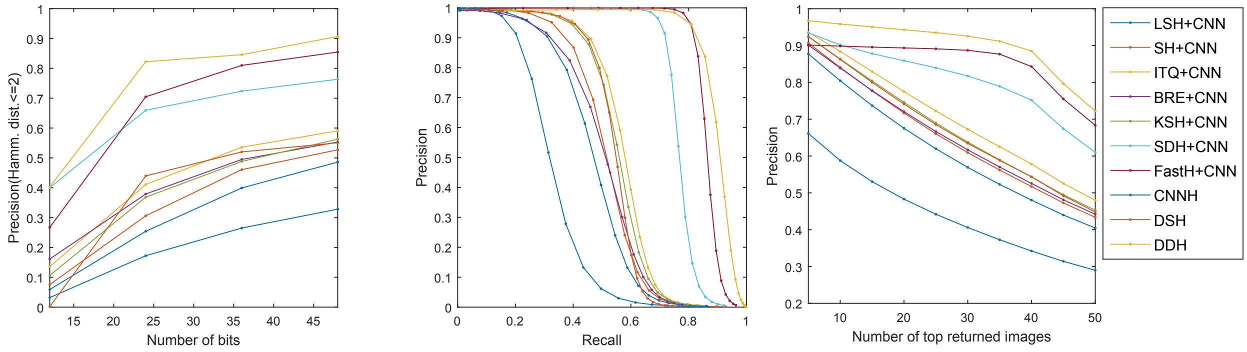
To compare the performance of hashing methods, we utilize four evaluation metrics: Mean Average Precision (MAP), Precision-Recall curves, Precision curves within Hamming distance 2 and Precision curves w.r.t different numbers of top returned samples.

4.2 Experimental Setting

To shown the effectiveness of the proposed DDH method, we compare it with 7 shallow hashing methods and two deep hashing methods. The shallow hashing methods are LSH [Andoni and Indyk, 2006], SH [Weiss *et al.*, 2008], ITQ [Gong *et al.*, 2011], KSH [Liu *et al.*, 2012], BRE [Kulis and Darrell, 2009], SDH [Shen *et al.*, 2015] and FastH [Lin *et al.*, 2014], while the deep methods are CNNH [Xia *et al.*, 2014] and DSH [Liu *et al.*, 2016]. For the shallow methods, the 236-dimensional LBP feature vector is utilized to represent the visual content of face images for both YouTube Faces and FaceScrub datasets. For deep methods, the original pixels are taken as the input. For fair comparison, the CNN

Table 1: The compared results in terms of Mean Average Precision (MAP) of different hashing methods on the YouTube Faces and FaceScrub datasets.

Method	YouTube Faces				FaceScrub			
	12 bits	24 bits	36 bits	48 bits	12 bits	24 bits	36 bits	48 bits
LSH	0.0094	0.0274	0.0491	0.0773	0.0027	0.0031	0.0033	0.0038
SH	0.0164	0.0520	0.0702	0.0990	0.0030	0.0034	0.0037	0.0038
ITQ	0.0198	0.0593	0.0986	0.1372	0.0033	0.0038	0.0042	0.0046
BRE	0.0277	0.0609	0.0954	0.1160	0.0031	0.0032	0.0033	0.0037
KSH	0.0174	0.0601	0.0960	0.1359	0.0031	0.0036	0.0035	0.0038
SDH	0.0317	0.0964	0.1346	0.1613	0.0028	0.0035	0.0041	0.0049
FastH	0.0877	0.2199	0.2983	0.3495	0.0048	0.0078	0.0113	0.0172
LSH + CNN	0.0590	0.2541	0.3993	0.4861	0.0092	0.0221	0.0333	0.0444
SH + CNN	0.0756	0.3060	0.4607	0.5271	0.0183	0.0298	0.0349	0.0395
ITQ + CNN	0.1357	0.4109	0.5358	0.5906	0.0258	0.0481	0.0683	0.0825
BRE + CNN	0.1613	0.3791	0.4947	0.5533	0.0210	0.0368	0.0498	0.0621
KSH + CNN	0.1062	0.3685	0.4881	0.5628	0.0217	0.0414	0.0596	0.0725
SDH + CNN	0.4003	0.6596	0.7237	0.7634	0.0191	0.0602	0.0938	0.1279
FastH + CNN	0.2671	0.7048	0.8099	0.8546	0.0373	0.0831	0.1274	0.1681
CNNH	0.0320	0.1723	0.2650	0.3283	N/A	N/A	N/A	N/A
DSH	0.0019	0.4396	0.5201	0.5507	0.0046	0.0062	0.0075	0.0081
DDH	**0.4029**	**0.8223**	**0.8457**	**0.9068**	**0.0650**	**0.1103**	**0.1437**	**0.1889**



(a) (b) (c)

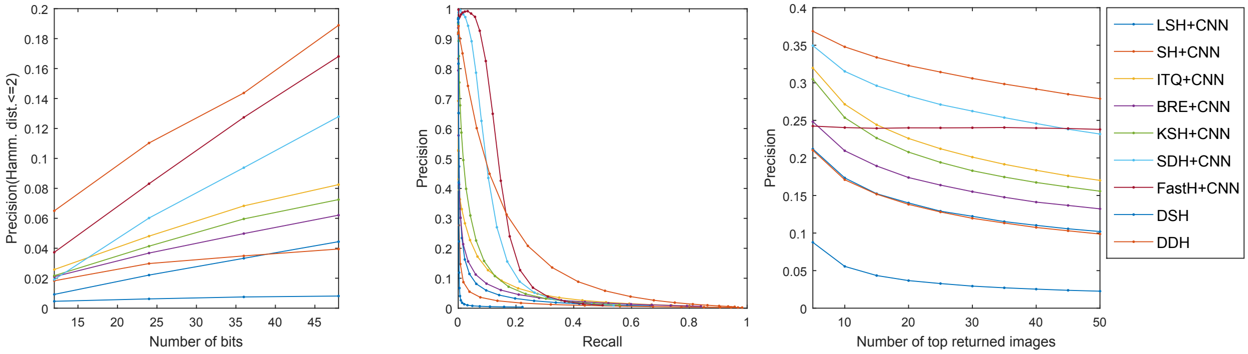
Figure 2: The results on the YouTube Faces dataset. (a) Precision curves with Hamming radius 2; (b) Precision-recall curves of Hamming ranking with 48 bits; (c) Precision curves with 48 bits w.r.t. the number of top returned samples.

features are extracted and used as the input of all the shallow hashing methods, which are denoted as "+CNN", such as "LSH+CNN" and "FastH+CNN".

For the proposed deep architecture, there are some details to configured in advance. The filter size in first convolutional layer is 3×3 with stride 1, and the other convolutional layers employ the filter with the size of 2×2 with stride 1. Weights in the fourth convolutional layer are totally unshared within the region of 2×2. The numbers of feature maps of the four convolutional layers are set to 20, 40, 60 and 80 respectively. For the activation function, all convolutional layers and the Face Feature layer use Rectification Linear Unit (ReLU) [Krizhevsky *et al.*, 2012]. For the parameter setting of the divide-and-encode module, the number of each group on the Slice layer is set to 4 in our experiments. In addition, we apply Batch Normalization [Ioffe and Szegedy, 2015] after each convolutional layer to accelerate the convergence speed of the objective function. Furthermore, we adopt Adam [Kingma and Ba, 2014] as our optimization algorithm and fix the batch size as 256 during training. We use a training set of 10 percent as a validation set to verify the results of the training set. For the parameters α and β, in experiments we set them to 0.0002 and 1, respectively. All the experiments are implemented with Keras framework on a NVIDIA GTX 850M with CUDA7.5 and cuDNN v5.1.

4.3 Results and Discussions

Experiments are conducted on the YouTube Faces and FaceScrub datasets, and the performance is first evaluated in terms of MAP. The results are presented in Table 1. The length of hash codes varies from 12 to 48. From the results in Table 1, we can have the following observations. First, the proposed DDH method achieves the best results and are significantly superior to other hashing methods. By introducing the divide-and-encode module, more compact codes can be learned and the size of parameters on the Slice layer become $1/K$ of the



(a) (b) (c)

Figure 3: The results on the Facescrub dataset. (a) Precision curves with Hamming radius 2; (b) Precision-recall curves of Hamming ranking with 48 bits; (c) Precision curves with 48 bits w.r.t. the number of top returned samples.

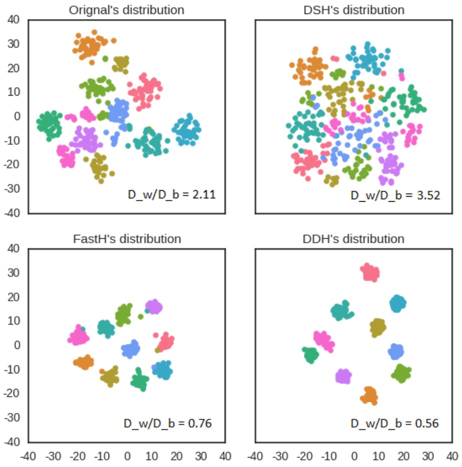


Figure 4: Visualization of the distributions of hash codes on the YouTube Faces dataset ($K = 48$). D_w and D_b are the inter-class and intra-class variations, respectively. Best viewed in color.

the size of parameters with the full-connected layer. Second, compared with CNNH and DSH, DDH obtains much improvement by integrating the hashing learning and prediction learning into one unified framework. And the results of DDH with shorter hash codes are better than ones of CNNH and DSH with longer hash codes in some cases, which indicates that DDH can learn compact hash codes. Third, deep hashing methods are all better than the shallow hashing methods and the shallow hashing methods with CNN features. It demonstrates that the advantages of the ene-to-end learning by incorporating the deep structure and hash learning into one framework. Furthermore, the shallow hashing methods with CNN features yield better result than them with hand-crafted features, which is consistent with many previous works. Finally, the proposed DDH method can learn discriminative and compact hash codes by integrating the feature extraction, the divide-and-encode module, the hashing learning and prediction learning into one unified deep network.

Besides, the performance of hashing methods is evaluated in terms of Precision-Recall curves, Precision curves within Hamming distance 2 and Precision curves w.r.t the number of top returned samples. Figure 2 and Figure 3 show the results on the YouTube Faces and FaceScrub datasets, respectively. Since the shallow hashing methods with CNN features are better than them with hand-crafted features, only the results of the shallow hashing methods with CNN features are reported. From the results, we can obtain the same observations as ones from Table 1. The proposed DDH method is significantly better than other hashing methods. That is because the proposed method can generate discriminative and compact binary hash codes of faces for face image retrieval. The proposed network integrates feature extraction and hash learning into a unified framework, which can guarantee that the learned features are compatible with the hashing learning.

To better illustrate the discriminative power of DDH, the distributions of the learned hash codes with 48 bits by different deep hashing methods are visualized by using t-SNE [Laurens and Hinton, 2008], and the results on the YouTube Faces dataset are shown in Figure **??**. It can be observed that the learned hash codes by the proposed DDH method have larger extra-class variation and smaller intra-class variation simultaneously. That is, the learned hash codes by DDH are more discriminative. Face images belonging to the same class can be easily deemed as similar for image retrieval.

5 Conclusion

In this work, we propose a novel Discriminative Deep Hashing (DDH) framework for large-scale face image retrieval. The proposed framework leverages feature extraction, hash learning and class prediction in one unified network. And the divide-and-encode module is introduced to reduce the redundancy among hash codes and the network parameters simultaneously. The proposed method can learn discriminative and compact hash codes. Experiments show that the proposed method achieves encouraging retrieval performance.

Acknowledgments

This work was partially supported by the 973 Program of China (Project No. 2014CB347600), National Natural Science Foundation of China (Grant No. 61522203 and 61402228) and Natural Science Foundation of Jiangsu Province under Grant BK20140058.

References

[Andoni and Indyk, 2006] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *IEEE Symposium on Foundations of Computer Science*, 2006.

[Bergstra *et al.*, 2010] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: A cpu and gpu math expression compiler. In *Python in Science Conference*, 2010.

[Dong *et al.*, 2016] Zhen Dong, Su Jia, Tianfu Wu, and Mingtao Pei. Face video retrieval via deep learning of binary hash representations. In *AAAI*, 2016.

[Gong *et al.*, 2011] 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*, 35(12):2916–29, 2011.

[Ioffe and Szegedy, 2015] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[Kingma and Ba, 2014] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

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

[Kulis and Darrell, 2009] Brian Kulis and Trevor Darrell. Learning to hash with binary reconstructive embeddings. In *NIPS*, 2009.

[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.

[Laurens and Hinton, 2008] Van Der Maaten Laurens and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9(2605):2579–2605, 2008.

[Li *et al.*, 2015] Zechao Li, Jing Liu, Jinhui Tang, and Hanqing Lu. Robust structured subspace learning for data representation. *IEEE TPAMI*, 37(10):2085–2098, 2015.

[Lin *et al.*, 2013] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *CoRR*, abs/1312.4400, 2013.

[Lin *et al.*, 2014] Guosheng Lin, Chunhua Shen, Qinfeng Shi, Anton Van Den Hengel, and David Suter. Fast supervised hashing with decision trees for high-dimensional data. In *CVPR*, 2014.

[Lin *et al.*, 2015] Kevin Lin, Huei-Fang Yang, Jen-Hao Hsiao, and Chu-Song Chen. Deep learning of binary hash codes for fast image retrieval. In *CVPR*, 2015.

[Liu *et al.*, 2012] Wei Liu, Jun Wang, Rongrong Ji, and Yu Gang Jiang. 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.

[Ng and Winkler, 2014] Hongwei Ng and Stefan Winkler. A data-driven approach to cleaning large face datasets. In *ICIP*, 2014.

[Ojala *et al.*, 2002] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE TPAMI*, 24(7):971–987, 2002.

[Schroff *et al.*, 2015] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

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

[Sun *et al.*, 2014a] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. In *NIPS*, 2014.

[Sun *et al.*, 2014b] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. In *CVPR*, 2014.

[Taigman *et al.*, 2014] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, 2014.

[Tang *et al.*, 2015] Jinhui Tang, Zechao Li, Meng Wang, and Ruizhen Zhao. Neighborhood discriminant hashing for large-scale image retrieval. *IEEE TIP*, 24(9):2827–2840, 2015.

[Tang *et al.*, 2017] Jinhui Tang, Zechao Li, and Xiang Zhu. Supervised deep hashing for scalable face image retrieval. *Pattern Recognition*, 2017. Online.

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

[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.

[Wolf *et al.*, 2011] Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In *CVPR*, 2011.

[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.

[Zhao *et al.*, 2015] Fang Zhao, Yongzhen Huang, Liang Wang, and Tieniu Tan. Deep semantic ranking based hashing for multi-label image retrieval. In *CVPR*, 2015.

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