# Face Video Retrieval via Deep Learning of Binary Hash Representations

**Zhen Dong**[1] and **Su Jia**[2] and **Tianfu Wu**[3,4] and **Mingtao Pei**[1]*
1. Beijing Laboratory of IIT, School of Computer Science, Beijing Institute of Technology, Beijing, China
2. Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, USA
3. Institute of Sensing Technology, Beijing University of Posts and Telecommunications, Beijing, China
4. Department of Statistics, University of California, Los Angeles, USA

## Abstract

Retrieving faces from large mess of videos is an attractive research topic with wide range of applications. Its challenging problems are large intra-class variations, and tremendous time and space complexity. In this paper, we develop a new deep convolutional neural network (deep CNN) to learn discriminative and compact binary representations of faces for face video retrieval. The network integrates feature extraction and hash learning into a unified optimization framework for the optimal compatibility of feature extractor and hash functions. In order to better initialize the network, the low-rank discriminative binary hashing is proposed to pre-learn hash functions during the training procedure. Our method achieves excellent performances on two challenging TV-Series datasets.

## Introduction

Given a face video of a person, face video retrieval aims to search videos containing the person from the video database. Face video retrieval is an attractive research area with wide range of applications, such as locating and tracking a criminal suspect from surveillance videos, retrieving masses of long videos to annotate face data for vision researchers, collecting all videos of a person from videos captured by his/her family digit camera, and the intelligent fast-forward of movies.

The challenging problems of face video retrieval are large intra-class variations of faces, and the strong demands of time and space saving. The faces in Figure 1 show the dramatic intra-class variations caused by poses, lighting conditions, expressions, clothes, background interferences, and the orientation of the actor in TV-Series, which indicates that good representations of faces should be robust to these variations and discriminative between classes. Moreover, the representations have to be compact for fast retrieval and space saving. Existing video based face recognition methods (Huang et al. 2015; Chen et al. 2013) represent face videos by features with thousands of floating point numbers or more, resulting in tremendous time and space complexity and being unapplicable to the face video retrieval task. To solve these problems, we propose to learn discriminative and compact representations of faces by deep convolutional neural network (deep CNN) for face retrieval in videos.

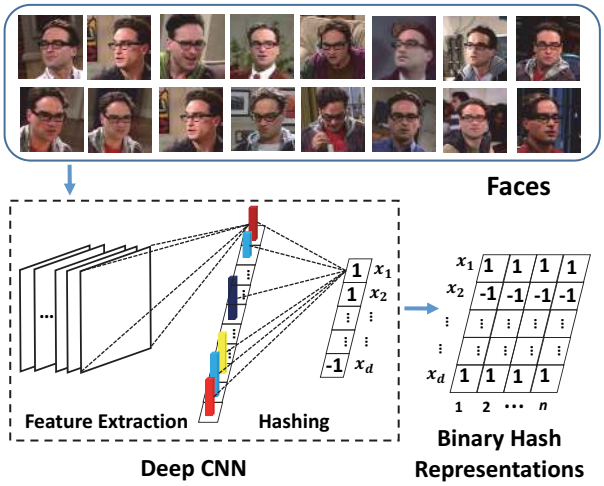


Figure 1: The hash learning is integrated into the deep CNN to obtain discriminative and compact representations for face video retrieval. An actor (*Leonard Hofstadter*) in TV-Series (*the BigBang Theory*) has dramatic intra-class variations mainly caused by pose, lighting, expression, clothes, background interferences, and the orientation of the actor. The deep CNN is able to generate similar representations for faces with large intra-variations.

Recently, deep CNN has been successfully applied on many vision tasks, such as image classification (Krizhevsky, Sutskever, and Hinton 2012) and image segmentation (Sharma, Tuzel, and Jacobs 2015), which shows the powerful ability of deep CNN for describing complex non-linear mappings and learning rich mid-level representations. The CNN features learned from data are discriminative, but still high dimensional for the retrieval task. In retrieval tasks, the hashing-based methods which project high dimensional features to a low dimensional space are often used (Liu et al. 2012; Li et al. 2015b), but feature extraction and hash function learning in these methods are independent, in other words, the extracted features may not be compatible with the hash coding process. Therefore, we fuse the hash learn-

*corresponding author

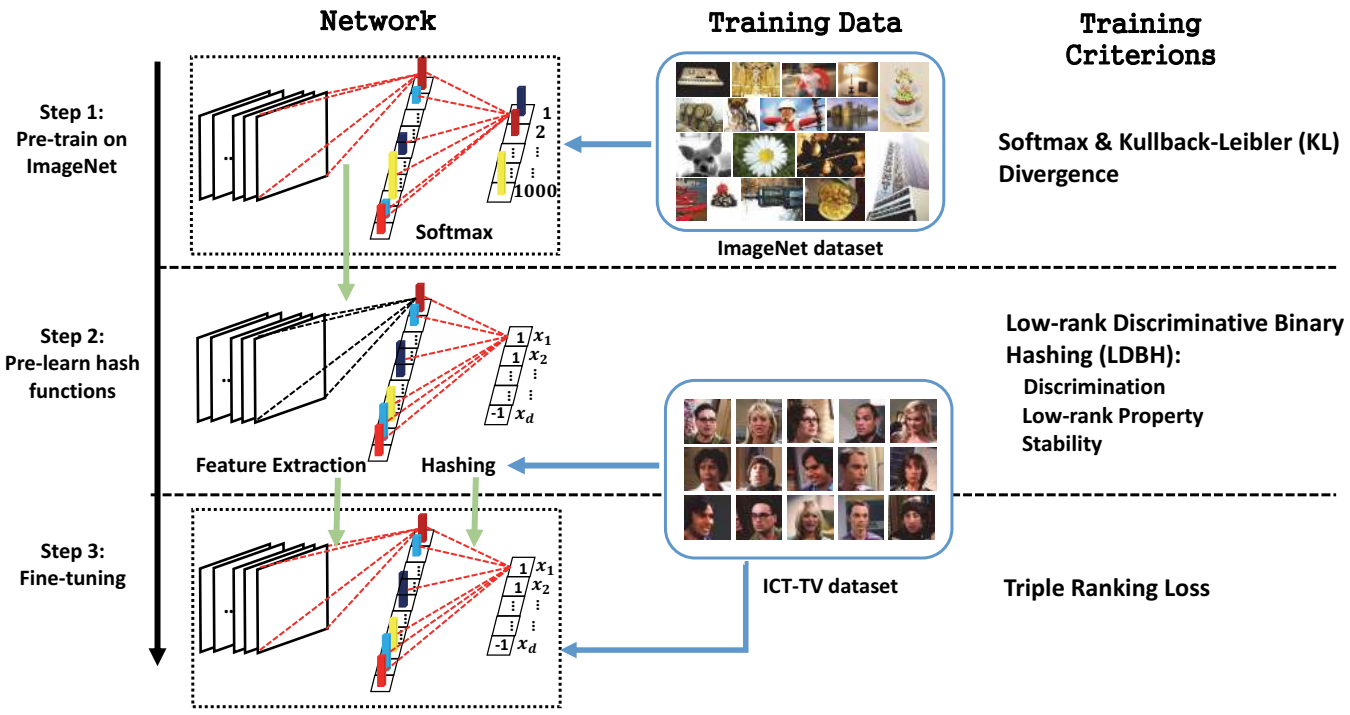


Figure 2: The training procedure of the deep CNN contains three steps. First, we use the front seven layers of the well-known AlexNet which is trained on the large-scale ImageNet dataset as the feature extractor of the network for good initializations. Second, the low-rank discriminative binary hashing is proposed to learn hash functions which serve as the last layer of the network. The proposed hash learning method takes the discrimination, low-rank property, and stability into account to better initialize the last layer. At last, the fine-tuning procedure is conducted with large amounts of face images. The red dotted lines represent parameters which can be learned in the corresponding step, the green arrow means parameter transfering, and the blue arrow shows the training data used in each step.

ing into the deep CNN and build an end-to-end system to learn discriminative and compact representations of faces as shown in Figure 1.

Figure 2 shows that our deep convolutional neural network is trained in three steps. In the first step, the network is trained on the large-scale ImageNet dataset (Krizhevsky, Sutskever, and Hinton 2012) to obtain good initializations. For convenience, the front seven layers of the well-known AlexNet are used as the feature extractor of our network. In the second step, we propose the low-rank discriminative binary hashing method to learn hash functions with large amounts of face images. To alleviate the problem of large intra-class variations of faces, the hash functions are learned under the guide of supervision information to ensure that samples belonging to the same class have similar hash codes. In the third step, the network is specifically fine-tuned for face video retrieval. The objective of this step is to separate the positive sample pair (a pair of samples coming from the same class) from the negative pair by a distance margin, which is quite effective in retrieval system (Lai et al. 2015; Zhao et al. 2015; Norouzi, Blei, and Salakhutdinov 2012). The network takes a face image as input and outputs the corresponding compact binary hash representation where each bit can be regarded as a visual attribute of the face. We conduct experiments on two challenging TV-Series datasets (*the Big Bang Theory* and *the Prison Break*) (Li et al. 2015c) and achieve good performances.

To the best of our knowledge, this is the first paper that uses deep neural network on face video retrieval. The contributions of the paper are three-folds: (1) The proposed deep CNN is able to generate discriminative and compact binary representations of faces for face video retrieval. The network integrates the feature extraction and hash learning into a unified optimization framework to ensure that the extracted features are compatible with the hashing procedure. (2) The proposed low-rank discriminative binary hashing takes the discrimination, low-rank property, and the stability into account, and shows its effectiveness compared with the state-of-the-art hash learning methods. (3) The proposed method achieves excellent results of face video retrieval on two challenging TV-Series datasets.

The remainder of the paper is organized as follow. We first review the related work including face video retrieval methods, deep learning on face-related tasks, and hashing methods. Next, we elaborate the training procedures of the deep CNN. We then present the preliminary experiments of the proposed method on two TV-Series datasets, and end with conclusions.

## Related Work

In this section, we give a brief review of previous literatures related to our work including face video retrieval, deep learning on face-related tasks, and hashing methods.

### Face Video Retrieval

There are few works on face video retrieval. Arandjelović and Zisserman (2005; 2006) introduced a film shot retrieval system based on face recognition algorithms. They represented the face shot as an identity preserving and variation insensitive signature image. Sivic, Everingham, and Zisserman (2005) developed a face video retrieval system which models the face shots as distributions of frames. The key point of these works is to build a complete retrieval system including the procedures of shot boundary detection, face tracking, etc., and the representations of face shots are high dimensional, which still face the problem of large intra-class variations and time and space cost. Different from these works, this paper aims to learn discriminative and compact binary hash representations of faces to improve the retrieval performance.

Li et al. () proposed a video coding method called compact video code (CVC) for face video retrieval in TV-Series. In their method, a face video is represented by its covariance matrix of frames' DCT features, and the CVC is used to obtain the binary codes of the face video. They (Li et al. 2015c) further represented a face video as the covariance matrix of Fisher Vector features, and executed the CVC in the Reproducing Kernel Hilbert Space. In the literature, two large scale TV-Series face video datasets are released where the image of each frame is provided rather than only providing low-level features. In (Li et al. 2015b), the face videos are retrieved with an image query. To measure the similarity of image and video, a new hashing method across Euclidean Space and Riemannian Manifold is proposed. Although these works achieve good performances, the feature extraction (DCT, Fisher Vector, and Covariance Matrix) and the hash function learning are independent, which causes the problem that extracted features might be incompatible with the hash coding process. In contrast, our deep CNN integrates the feature extraction and hash learning into a unified optimization framework where two procedures interact to generate optimal hash codes. Besides, since the network takes a face image as input, either an image or a video is available to be used as the query.

### Deep Learning on Face-related Tasks

Recently, deep learning has shown its promising performances on face-related tasks. Zhang et al. (2014) used a sequence of stacked auto-encoder networks in a coarse-to-fine architecture to infer face shapes from face images. Taigman et al. (2014; 2015) addressed the face verification problem by a nine-layer deep neural network with 120 million parameters, and reduced the error of the state-of-the-art by more than $25\%$ on the LFW dataset. Hu, Lu, and Tan (2014) presented a discriminative deep metric learning method for face verification by training a deep neural network which project face pairs into the same feature space where the distances between positive face pairs are less than a threshold. Liu et al. (2015) cascaded two CNNs to localize faces and learn attributes jointly, and improve existing methods 8 and 13 percents on the CelebFaces and LFW datasets, respectively. Sun, Wang, and Tang (2015) proposed a deep network named "DeepID2+" which achieves exciting results on both LFW and YouTube Faces datasets for both face identification and verification. Li et al. (2015a) presented a cascade CNN for fast face detection and obtained good performance on the "annotated faces in the wild" and the "face detection data set and benchmark". Schroff, Kalenichenko, and Philbin (2015) trained a deep CNN called FaceNet for face recognition and clustering. The FaceNet maps face image to a compact representation in Euclidean space and achieves a new record accuracy on both LFW and YouTube Faces datasets. These encouraging achievements of deep learning on face-related tasks move us towards higher performance in the application of face video retrieval. Different from above networks, our network outputs compact binary representations for the retrieval task.

### Hashing Methods

Hashing methods are widely used in retrieval systems owning to its encouraging efficiency in both speed and storage. As an representative of data-independent hashing methods, the locality sensitive hashing (LSH) (Gionis et al. 1999) uses random projections as hash functions. Despite theoretical asymptotic guarantees, LSH still needs long hash codes to get satisfactory retrieval results. In contrast, data-dependent methods, namely learning-based methods, aim to generate compact similarity-preserving hash codes by exploiting the structure or supervision information of the training data. The data-dependent methods can be roughly categorized as unsupervised, semi-supervised, and supervise methods. Unsupervised methods learn hash functions only by unlabeled training data, and the representatives are kernelized locality-sensitive hashing (KLSH) (Kulis and Grauman 2009), spectral hashing (SH) (Weiss, Torralba, and Fergus 2009), iterative quantization hashing (ITQ) (Gong and Lazebnik 2011), and Anchor graph hashing (AGH) (Liu et al. 2011). Semi-supervised and supervised methods improve the hash code quality by using supervision information, and the notable examples are discriminative binary coding (DBC) (Rastegari, Farhadi, and Forsyth 2012), semi-supervised hashing (SSH) (Wang, Kumar, and Chang 2010), kernel-based supervised hashing (KSH) (Liu et al. 2012), minimal loss hashing (MLH) (Norouzi and Blei 2011), and supervised iterative quantization hashing (SITQ) (Gong and Lazebnik 2011), etc.

Recently, several deep neural networks have been proposed to learn hash functions for specific vision tasks. Zhao et al. (2015) learned hash functions by a deep CNN in multilevel semantic ranking supervision manner for multi-label image retrieval. Lai et al. (2015) developed a supervised hash learning method based on the well known network, "network in network" (Min, Qiang, and Shuicheng 2014), for image retrieval. Lin et al. (2015) presented a rapid clothing retrieval system based on the AlexNet where a latent layer is added to generate hash codes. Since the deep network learns features and hash functions in a unified end-

to-end optimization form, these works achieve encouraging retrieval performances. Similar to these works, we present a deep CNN to learn binary hash representations of faces for face video retrieval.

## Approach

In this section, we elaborate the training procedures of the deep CNN. Figure 2 shows the three steps of the training procedures: pre-train the deep CNN, pre-learn hash functions, and fine-tuning.

### Pre-train on ImageNet Dataset

The deep CNN is pre-trained on the ImageNet dataset which has more than $1.2$ million images of $1000$ categories to obtain good initializations. We use the released AlexNet (Krizhevsky, Sutskever, and Hinton 2012) model which is trained on ImageNet dataset for convenience. The AlexNet contains convolutional layers, normalization layers, linear layers, ReLU activation layers, and max-pooling layers. For simplicity, we use $\mathbf{L}_{1-5}$ to represent the $5$ convolutional layers, and $\mathbf{L}_{6-8}$ describe the $3$ linear layers. The $\mathbf{L}_7$ outputs features with the dimension of $4096$, and the dimensionality of features in $\mathbf{L}_8$ is $1000$. The $\mathbf{L}_8$ is followed by a softmatx classifier to generate probability distribution for classification. Previous studies (Krizhevsky, Sutskever, and Hinton 2012; Oquab et al. 2014) show that the $4096$-dimensional features of $\mathbf{L}_7$ perform better than many handcrafted features. In our network, the $\mathbf{L}_{1-7}$ layers are used as the feature extractor.

### Pre-Learn Hash Functions

With the pre-trained deep CNN, rich mid-level features of faces are obtained. The features are high dimensional which results in tremendous time and space complexity. In order to generate effective and compact representations for retrieval, we propose the Low-rank Discriminative Binary Hashing (LDBH) to project the high dimensional CNN features to a much lower dimensional binary space. In the binary space, each bit of the representation is either $1$ or $-1$, so the distance between two representations can be easily computed by the XOR operation on bits, which reduces the time cost during the retrieval procedure. In addition, the low dimensionality of the binary space ensures the low space complexity.

Let $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_C] \in \mathbb{R}^{t \times n}$ be the obtained feature set of $C$ classes where $t$ and $n$ represent the dimension and number of training samples, respectively. Our goal is to learn hash functions $\mathbf{W}$ which project $\mathbf{X}$ from feature space into the binary space, *i.e.*

$$\mathbf{b}_i = sgn(\mathbf{W}^\top \mathbf{x}_i), \forall \mathbf{x}_i \in \mathbf{X}, \tag{1}$$

where $sgn(\cdot)$ is the sign function which returns $+1$ for positive number and returns $-1$ for negative number, and $\mathbf{b}_i$ is the binary code of $\mathbf{x}_i$. All the generated codes form a binary matrix $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_n] \in \{+1, -1\}^{s \times n}$ where $s \ll t$ is the dimension of the binary space, and $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_s] \in \mathbb{R}^{t \times s}$ contains $s$ hash functions.

To learn hash functions, three constraints are taken into account: discrimination, low-rank property, and stability. **First**, to alleviate the problem of large intra-class variations, hash codes in the binary space should be discriminative, in other words, samples of the same class should have similar codes, and samples of different classes can be better separated. The discriminative constraint is ensured by minimizing

$$f(\mathbf{B}) = \sum_{\substack{p=1, \\ \mathbf{b}_i, \mathbf{b}_j \in \mathbf{B}_p}}^{C} d(\mathbf{b}_i, \mathbf{b}_j) - \lambda \sum_{\substack{p=1, \\ \mathbf{b}_i \in \mathbf{B}_p}}^{C} \sum_{\substack{q=1, q \neq p, \\ \mathbf{b}_j \in \mathbf{B}_q}}^{C} d(\mathbf{b}_i, \mathbf{b}_j), \tag{2}$$

where $d(\cdot, \cdot)$ represents the Hamming distance of binary space, and $\lambda$ is the normalization parameter to balance the number of positive sample pairs and negative sample pairs. **Second**, the low-rank constraint of the binary code matrix is enforced. The low-rank constraint encourages the hash codes of the same class to be correlated, which reduces the redundancy of face video data. Since minimizing the rank of $\mathbf{B}$ is an NP-hard problem and difficult to solve, we use the the nuclear norm $\|\mathbf{B}\|_*$ , the convex envelope of the rank function, instead of the rank function during optimization. **Third**, the stability of the hash codes also needs to be considered. Each hash function can be viewed as a hyperplane in the feature space, and the stability means that the hyperplanes are largely marginalized like SVM. Therefore, we formulate the hash function learning as

$$\begin{aligned} \min_{\mathbf{W}, \xi, \mathbf{B}} & f(\mathbf{B}) + \eta \|\mathbf{B}\|_* + \frac{1}{2}\|\mathbf{W}\|_F^2 + \mu \sum_{i=1}^{d} \sum_{j=1}^{n} \xi_{ij} \\ s.t. \ & \mathbf{B}_{ij}(\mathbf{w}_i^\top \mathbf{x}_j) \geq 1 - \xi_{ij}, \\ & \xi_{ij} \geq 0, \\ & \mathbf{B} = sgn(\mathbf{W}^\top \mathbf{X}), \end{aligned} \tag{3}$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix, $\eta$ describes the importance of low-rank constraint, and $\mu$ is the parameter representing the trade-off between hashing error and capacity.

The objective function Eq.(3) is non-convex, so it's impractical to find the global minimum. Fortunately, the local minimum is able to generate effective hash codes in practice. The optimization procedures are formulated as follows: First, with the fixed $\mathbf{B}$, Eq.(3) w.r.t. $\mathbf{W}$ and $\xi$ is optimized by training $d$ linear SVMs where the $i$-th column of $\mathbf{B}$ and all columns of $\mathbf{X}$ are used as labels and training samples for the $i$-th SVM. Second, we compute the binary code matrix with the learned $\mathbf{W}$ as $\mathbf{B} = sgn(\mathbf{W}^\top \mathbf{X})$. Third, fixing $\mathbf{W}$ and $\xi$, Eq.(3) is optimized by introducing an auxiliary variable $\mathbf{A}$, and the objective function is formulated as

$$\begin{aligned} & \min_{\mathbf{A}, \mathbf{B}} f(\mathbf{B}) + \eta \|\mathbf{A}\|_* \\ & s.t. \ \mathbf{A} = \mathbf{B}, \mathbf{B} \in \{+1, -1\}^{s \times n}. \end{aligned} \tag{4}$$

We use the alternating direction method to optimize Eq.(4), and the augmented Lagrangian function is given by

$$f(\mathbf{B}) + \eta \|\mathbf{A}\|_* + tr(\mathbf{H}(\mathbf{A} - \mathbf{B})^\top) + \frac{\alpha}{2}\|\mathbf{A} - \mathbf{B}\|_F^2. \tag{5}$$

For fixed $\mathbf{B}$, the optimized $\mathbf{A}$ has closed solution (Cai, Candès, and Shen 2010), and $\mathbf{B}$ can be optimized by an efficient subgradient descend method proposed in (Rastegari, Farhadi, and Forsyth 2012). The algorithm of low-rank discriminative hashing is summarized in Algorithm 1.

**Algorithm 1:** Algorithm of low-rank discriminative binary hashing

**Input**: Feature set $\mathbf{X} \in \mathbb{R}^{t\times n}$.
**Output**: Hash functions $\mathbf{W} \in \mathbb{R}^{t\times d}$.

**repeat**
  Update $\mathbf{W}$ by training $d$ linear SVMs with $\mathbf{B}$ as labels;
  $\mathbf{A} = \mathbf{W}^\top\mathbf{X}, \mathbf{B} = sgn(\mathbf{A})$;
  **repeat**
    SVD decomposition: $\mathbf{B} - \mathbf{H}/\alpha = \mathbf{USV}^\top$;
    $\mathbf{A} = \mathbf{U}\big(sgn(\mathbf{S}) \otimes \max(|\mathbf{S} - \eta/\alpha|, 0)\big)\mathbf{V}^\top$;
    ($\otimes$ denotes the element-wise multiplication)
    Update $\mathbf{B}$ by subgradient descend method proposed in (Rastegari, Farhadi, and Forsyth 2012);
    $\mathbf{H} = \mathbf{H} + \alpha(\mathbf{A} - \mathbf{B}), \alpha = \gamma\alpha$;
  **until** *Converged*;
**until** *Converged*;

## Fine-tuning

The fine-tuning procedure integrates the feature extraction and hash function learning into a unified optimization framework and forms an end-to-end learning system. In the learning system, the feature extraction is optimally compatible with the hashing process, *i.e.*, the features are learned for hashing, and the performance of hash codes is able to guide the learning of face features. In this way, a pair of semantically similar faces will have similar compact hash representations, and the representations of faces from different class will have large distance. To this end, we use the triplet ranking loss which reflects the target we want in "face retrieval".

Suppose that the deep CNN describes the complex nonlinear mapping $g : \mathcal{I} \to \{+1, -1\}^s$ where $\mathcal{I}$ denotes the face image space, so a face image $\mathbf{q}$ is able to be represented as a $s$-bit binary representation $g(\mathbf{q})$. The triple ranking loss reflects the relative similarities in the form as "face $\mathbf{q}$ is more similar to $\widetilde{\mathbf{q}}$ than $\widehat{\mathbf{q}}$". Similar to (Norouzi, Blei, and Salakhutdinov 2012), the triple ranking loss for $((g(\mathbf{q}), g(\widetilde{\mathbf{q}}), g(\widehat{\mathbf{q}}))$ is defined as

$$\begin{aligned} &l\big(g(\mathbf{q}), g(\widetilde{\mathbf{q}}), g(\widehat{\mathbf{q}})\big) \\ &\quad = \max\Big(d\big(g(\mathbf{q}), g(\widetilde{\mathbf{q}})\big) - d\big(g(\mathbf{q}), g(\widehat{\mathbf{q}})\big) + \delta, 0\Big), \end{aligned} \tag{6}$$

where $d(\cdot,\cdot)$ is the Hamming distance, and $\delta \geq 0$ is a parameter controlling the margin of the distance difference. Define the training face image set as $\mathbf{Q} = [\mathbf{Q}_1, \mathbf{Q}_2, ..., \mathbf{Q}_C]$ of $C$ classes, the objective of fine-tuning the deep CNN is

$$\min_{\mathbf{W}^\star,\mathbf{W}} \sum_{i=1}^{C} \sum_{\substack{\mathbf{q},\widetilde{\mathbf{q}}\in\mathbf{Q}_i \\ \mathbf{q}\neq\widetilde{\mathbf{q}}}} \sum_{j\neq i, \widehat{\mathbf{q}}\in\mathbf{Q}_j} l\big(g(\mathbf{q}), g(\widetilde{\mathbf{q}}), g(\widehat{\mathbf{q}})\big), \tag{7}$$

where $\mathbf{W}$ is the parameters of the last layer (hash functions), and $\mathbf{W}^\star$ represents the parameters of the front layers.

To solve Eq.(7), the gradient of Eq.(6) is needed. Since the hash function contains the sign function $sgn(\cdot)$ which is non-smooth and non-differentiable, we use $\tanh(\cdot)$ instead of the sign function during the fine-tuning procedure. The hamming distance can be rewritten as $d\big(g(\mathbf{q}), g(\widetilde{\mathbf{q}})\big) = (s - g(\mathbf{q})^\top g(\widetilde{\mathbf{q}}))/2$. Therefore, the gradients of Eq.(6) w.r.t. hash codes are given by

$$\begin{aligned} &\frac{\partial l}{\partial g(\mathbf{q})} = \frac{1}{2}(g(\widehat{\mathbf{q}}) - g(\widetilde{\mathbf{q}})) \times I, \\ &\frac{\partial l}{\partial g(\widetilde{\mathbf{q}})} = -\frac{1}{2}g(\mathbf{q}) \times I, \quad \frac{\partial l}{\partial g(\widehat{\mathbf{q}})} = \frac{1}{2}g(\mathbf{q}) \times I, \end{aligned} \tag{8}$$

where $I$ is a binary function which returns 1 when $d\big(g(\mathbf{q}), g(\widetilde{\mathbf{q}})\big) - d\big(g(\mathbf{q}), g(\widehat{\mathbf{q}})\big) + \delta > 0$ and 0 for other occasions. Obtaining these gradients, the fine-tuning procedure can be conducted via the back-propagation algorithm.

# Experiments

## Dataset

We use the ICT-TV dataset (Li et al. 2015c) to evaluate the proposed method. The ICT-TV dataset contains two large scale video collections from two hit American shows, *i.e.*, the Big Bang Theory (BBT) and Prison Break (PB). These two TV series are quite different in their filming styles. The BBT is a sitcom with 5 main characters, and most scenes are taken indoors during each episode of about 20 minutes long. Differently, many shots of the PB are taken outside during the episodes with the length of about 42 minutes, which results in a large range of different illumination. All the face video shots are collected from the whole first season of both TV series, *i.e.*, 17 episodes of BBT, and 22 episodes of PB, and the number of shots of the two sets are $4,667$ and $9,435$, respectively. The collected video shots are stored in the form of images with size of $150 \times 150$ frame by frame.

## Experimental Settings

We compare our method with eight state-of-the-art hashing methods: LSH (Gionis et al. 1999), SH (Weiss, Torralba, and Fergus 2009), ITQ, SITQ, RR (Gong and Lazebnik 2011), SSH (Wang, Kumar, and Chang 2010), DBC (Rastegari, Farhadi, and Forsyth 2012), KSH (Liu et al. 2012), and two face video retrieval methods: HSVBC and HHSVBC (Li et al. 2015c). For each TV-Series dataset, we randomly select 10 face shots per actor or actress for training hash functions, and use the rest face shots for testing. Same to (Li et al. 2015c), the query set consists of 10 face shots. To evaluate the quality of hashing, we use three evaluation criterions: the Mean Average Precision (MAP), the Precision Recall (PR) curve, and Precision curve w.r.t. different number of top returned samples. For fair comparisons, all the methods use the same training and test sets.

The deep CNN is trained by the stochastic gradient descent with momentum of 0.9 and weight decay of 0.0005. The mini-batch size of the training samples is 128, and the triplets are randomly generated based on the labels. Our network takes the face image as input and outputs the binary hash representation of the face, and each bit of the binary

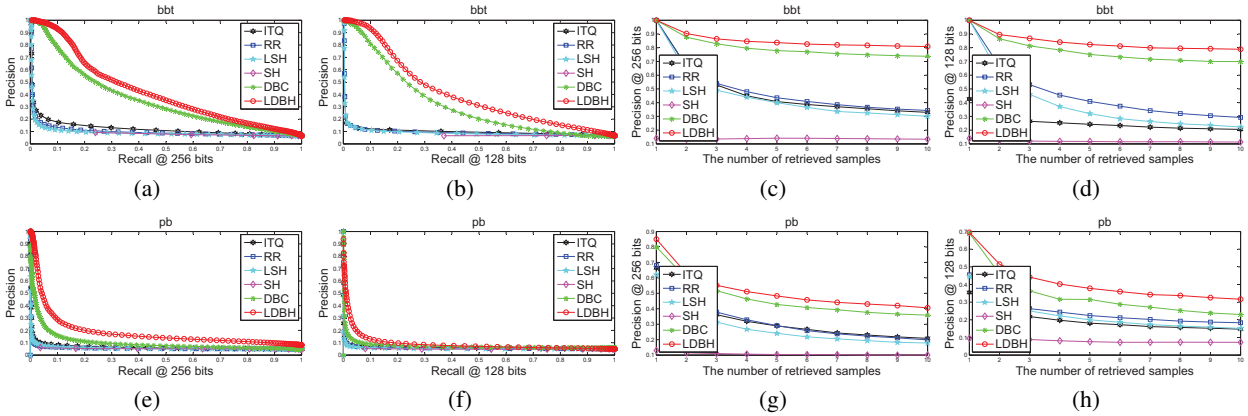


Figure 3: The comparison of PR curves and Precision curves w.r.t. the number of top returned samples (PN curve) between the LDBH and other hashing methods on two TV-Series datasets. (a) is the PR curve on BBT dataset with 256-dim hash code, (b) is the PR curve on BBT dataset with 128-dim hash code, (c) is the PN curve on BBT dataset with 256-dim hash code, (d) is the PN curve on BBT dataset with 128-dim hash code, (e) is the PR curve on PB dataset with 256-dim hash code, (f) is the PR curve on PB dataset with 128-dim hash code, (g) is the PN curve on PB dataset with 256-dim hash code, and (h) is the PN curve on PB dataset with 128-dim hash code.

Table 1: Comparison results with other retrieval methods on two datasets.

| Methods | *the Big Bang Theory* | | | | | | *Prison Break* | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 8 bits | 16 bits | 32 bits | 64 bits | 128 bits | 256 bits | 8 bits | 16 bits | 32 bits | 64 bits | 128 bits | 256 bits |
| LSH | 0.3533 | 0.3783 | 0.4093 | 0.4148 | 0.4414 | 0.4383 | 0.0959 | 0.0998 | 0.1048 | 0.1081 | 0.1078 | 0.1101 |
| RR | 0.3885 | 0.4207 | 0.4042 | 0.4507 | 0.4622 | 0.4407 | 0.0985 | 0.0981 | 0.1018 | 0.1042 | 0.1065 | 0.1105 |
| ITQ | 0.3434 | 0.3445 | 0.4033 | 0.4257 | 0.4428 | 0.4324 | 0.0999 | 0.1129 | 0.1083 | 0.1114 | 0.1095 | 0.1098 |
| SH | 0.4086 | 0.4225 | 0.3802 | 0.3809 | 0.3765 | 0.3972 | 0.0914 | 0.0901 | 0.0978 | 0.0964 | 0.1048 | 0.1059 |
| SSH | 0.3401 | 0.3134 | 0.2830 | 0.2757 | 0.2878 | 0.3656 | 0.1138 | 0.1527 | 0.1488 | 0.1417 | 0.1409 | 0.1436 |
| KSH | 0.4981 | 0.5799 | 0.6506 | 0.6965 | 0.7094 | 0.7300 | 0.1218 | 0.1571 | 0.1546 | 0.1619 | 0.1630 | 0.1599 |
| SITQ | 0.5384 | 0.6185 | 0.6702 | 0.6891 | 0.7006 | 0.7165 | 0.1070 | 0.1211 | 0.1326 | 0.1462 | 0.1578 | 0.1640 |
| HSVBC | 0.6208 | 0.7918 | 0.8494 | 0.8573 | 0.8663 | 0.8730 | 0.1393 | 0.1552 | 0.1824 | 0.1996 | 0.2158 | 0.2365 |
| HHSVBC | 0.7177 | 0.8763 | 0.9113 | 0.9078 | 0.9116 | 0.9172 | 0.1703 | 0.1950 | 0.2279 | 0.2585 | 0.2743 | 0.3035 |
| Our method | 0.7637 | 0.9171 | 0.9246 | 0.9317 | 0.9381 | 0.9412 | 0.1913 | 0.2107 | 0.2452 | 0.2778 | 0.2926 | 0.3261 |

representation can be viewed as a visual attribute of the face. We model the face video as a set of face images. Given a face video, each frame is inputted into the deep CNN to obtain a binary representation. All the binary representations are fused by hard-voting method, and a unified binary representation of the video is obtained for retrieval.

## Results and Discussions

We first evaluate the effectiveness of the proposed Low-rank Discriminative Binary Hashing (LDBH) method. The 4096-dim AlexNet features of face images are used to learn hash representations, and Figure 3 shows the comparison of PR curves and Precision curve w.r.t. different number of top returned samples between the LDBH and several state-of-the-art methods on two datasets. The two curves are computed by averaging the randomly selected $10 \times C$ queries where $C$ is the number of the character. Among these hashing methods, ITQ, RR, LSH, and SH are unsupervised methods, and the DBC and LDBH are supervised methods. As shown in the figure, the performances of supervised methods are much better than unsupervised methods due to the full usage of the supervision information. The primal reason of the fact that the LDBH achieves better results than DBC is that the LDBH considers low-rank property besides discrimination and stability.

After fine-tuning, the network is adjusted to generate discriminative and compact binary representation for retrieving faces from videos. Table 1 shows the comparisons of mAPs between the proposed method and other methods. The results of other methods are from (Li et al. 2015c). All these methods use the covariance matrix of Fisher Vectors as features which are not optimal compatible with the hashing procedure. In contrast, our method integrates feature extraction and hash learning into a unified network and achieves good performance.

## Conclusions

This paper presented a deep CNN for face video retrieval. The network integrates feature extraction and hash learning into a unified optimization framework to guarantee that the feature extractor is optimally compatible with the followed hashing. The low-rank discriminative binary hashing is proposed to pre-learn hash functions for better initializing the network, and the feature extractor is initialized by the front

seven layers of the AlexNet which is pre-trained on the ImageNet dataset. After obtaining the initializations, the network is fine-tuned to improve the performance of face video retrieval. The proposed method achieved excellent performances on two challenging TV-Series datasets.

## Acknowledgments

This work was supported in part by China 973 Program under grant No. 2012CB316300, and the Natural Science Foundation of China (NSFC) under grant No. 61472038.

## References

Arandjelović, O., and Zisserman, A. 2005. Automatic face recognition for film character retrieval in feature-length films. In *CVPR*, volume 1, 860–867. IEEE.

Arandjelović, O., and Zisserman, A. 2006. On film character retrieval in feature-length films. In *Interactive Video*. Springer. 89–105.

Cai, J.-F.; Candès, E. J.; and Shen, Z. 2010. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization* 20(4):1956–1982.

Chen, Y.-C.; Patel, V. M.; Shekhar, S.; Chellappa, R.; and Phillips, P. J. 2013. Video-based face recognition via joint sparse representation. In *FG*, 1–8. IEEE.

Gionis, A.; Indyk, P.; Motwani, R.; et al. 1999. Similarity search in high dimensions via hashing. In *VLDB*, volume 99, 518–529.

Gong, Y., and Lazebnik, S. 2011. Iterative quantization: A procrustean approach to learning binary codes. In *CVPR*, 817–824. IEEE.

Hu, J.; Lu, J.; and Tan, Y.-P. 2014. Discriminative deep metric learning for face verification in the wild. In *CVPR*, 1875–1882. IEEE.

Huang, Z.; Wang, R.; Shan, S.; and Chen, X. 2015. Projection metric learning on grassmann manifold with application to video based face recognition. In *CVPR*, 140–149. IEEE.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*, 1097–1105.

Kulis, B., and Grauman, K. 2009. Kernelized locality-sensitive hashing for scalable image search. In *ICCV*, 2130–2137. IEEE.

Lai, H.; Pan, Y.; Liu, Y.; and Yan, S. 2015. Simultaneous feature learning and hash coding with deep neural networks. In *CVPR*, 3270–3278. IEEE.

Li, Y.; Wang, R.; Cui, Z.; Shan, S.; and Chen, X. Compact video code and its application to robust face retrieval in tv-series. In *BMVC*. BMVA Press.

Li, H.; Lin, Z.; Shen, X.; Brandt, J.; and Hua, G. 2015a. A convolutional neural network cascade for face detection. In *CVPR*, 5325–5334. IEEE.

Li, Y.; Wang, R.; Huang, Z.; Shan, S.; and Chen, X. 2015b. Face video retrieval with image query via hashing across euclidean space and riemannian manifold. In *CVPR*, 4758–4767. IEEE.

Li, Y.; Wang, R.; Shan, S.; and Chen, X. 2015c. Hierarchical hybrid statistic based video binary code and its application to face retrieval in tv-series. In *FG*, 1–8. IEEE.

Lin, K.; Yang, H.-F.; Liu, K.-H.; Hsiao, J.-H.; and Chen, C.-S. 2015. Rapid clothing retrieval via deep learning of binary codes and hierarchical search. In *ICMR*, 499–502. ACM.

Liu, W.; Wang, J.; Kumar, S.; and Chang, S.-F. 2011. Hashing with graphs. In *ICML*, 1–8.

Liu, W.; Wang, J.; Ji, R.; Jiang, Y.-G.; and Chang, S.-F. 2012. Supervised hashing with kernels. In *CVPR*, 2074–2081. IEEE.

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep learning face attributes in the wild. In *ICCV*. IEEE.

Min, L.; Qiang, C.; and Shuicheng, Y. 2014. Network in network. In *ICLR (arXiv:1409.1556)*.

Norouzi, M., and Blei, D. M. 2011. Minimal loss hashing for compact binary codes. In *ICML*, 353–360.

Norouzi, M.; Blei, D. M.; and Salakhutdinov, R. R. 2012. Hamming distance metric learning. In *NIPS*, 1061–1069.

Oquab, M.; Bottou, L.; Laptev, I.; and Sivic, J. 2014. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 1717–1724. IEEE.

Rastegari, M.; Farhadi, A.; and Forsyth, D. 2012. Attribute discovery via predictable discriminative binary codes. In *ECCV*. Springer. 876–889.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 815–823. IEEE.

Sharma, A.; Tuzel, O.; and Jacobs, D. W. 2015. Deep hierarchical parsing for semantic segmentation. In *CVPR*, 530–538. IEEE.

Sivic, J.; Everingham, M.; and Zisserman, A. 2005. Person spotting: video shot retrieval for face sets. In *Image and Video Retrieval*. Springer. 226–236.

Sun, Y.; Wang, X.; and Tang, X. 2015. Deeply learned face representations are sparse, selective, and robust. In *CVPR*, 2892–2900. IEEE.

Taigman, Y.; Yang, M.; Ranzato, M.; and Wolf, L. 2014. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, 1701–1708. IEEE.

Taigman, Y.; Yang, M.; Ranzato, M.; and Wolf, L. 2015. Web-scale training for face identification. In *CVPR*, 2476–2574. IEEE.

Wang, J.; Kumar, S.; and Chang, S.-F. 2010. Semi-supervised hashing for scalable image retrieval. In *CVPR*, 3424–3431. IEEE.

Weiss, Y.; Torralba, A.; and Fergus, R. 2009. Spectral hashing. In *NIPS*, 1753–1760.

Zhang, J.; Shan, S.; Kan, M.; and Chen, X. 2014. Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment. In *ECCV*. Springer. 1–16.

Zhao, F.; Huang, Y.; Wang, L.; and Tan, T. 2015. Deep semantic ranking based hashing for multi-label image retrieval. In *CVPR*, 1556–1564. IEEE.