Heterogeneous Hashing Network for Face Retrieval Across Image and Video Domains

Chenchen Jing, Zhen Dong, Mingtao Pei, and Yunde Jia, *Member, IEEE*

***Abstract*—In this paper, we present a heterogeneous hashing network to generate effective and compact hash representations of both face images and face videos for face retrieval across image and video domains. The network contains an image branch and a video branch to project face images and videos into a common space, respectively. Then, the non-linear hash functions are learned in the common space to obtain the corresponding binary hash representations. The network is trained with three loss functions: 1) the Fisher loss; 2) the softmax loss; and 3) the triplet ranking loss. The Fisher loss uses the difference form of within-class and between-class scatter and is appropriate for the mini-batch-based optimization method. The Fisher loss together with the softmax loss is exploited to enhance the discriminative power of the common space. The triplet ranking loss is enforced on the final binary hash representations to improve retrieval performance. Experiments on a large-scale face video dataset and two challenging TV-series datasets demonstrate the effectiveness of the proposed method.**

***Index Terms*—Face retrieval, image and video domains, deep CNN, hash learning.**

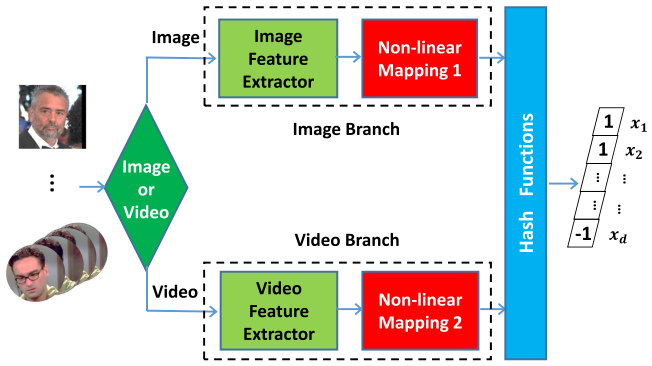


Fig. 1. Conceptual illustration of our heterogeneous hashing network. The input to the network can be either a face image or a face video. The face image is processed by the image branch, while the face video is processed by the video branch. Both branches have two modules: one for feature extraction and the other for mapping features from heterogeneous spaces into a common space. In the common space, hash functions are learned for the face retrieval task.

I. Introduction

FACE retrieval across image and video domains is a method to retrieve video shots of a person using his/her image (query-by-image video retrieval) or to retrieve face images using his/her video clip as a query (query-by-video image retrieval).

The "query-by-image video retrieval" task plays an important role in rapidly locating and tracking a person from surveillance video using the photo from an ID card, passport, or driver's license as the query. The "query-by-video image retrieval" task helps to determine the identity of an unknown person by retrieving a huge mug-shot image database and using his/her video shot taken by surveillance cameras as an input.

The core task of face retrieval across the image and video domains is to measure the similarity between a face image and a face video. A straightforward approach is to use the average or maximum of the similarities between the image and each frame of the video. Obviously, it neglects valuable information regarding the correlations between frames and also suffers from the problem of high computation and memory costs, particularly when the face video contains hundreds or thousands of frames. Modeling all the frames of a face video collectively is more promising, and many compact low-level video features are proposed, such as Gaussian distribution [1], [2], linear subspace descriptor [3]–[5], and SPD matrix descriptor [6]–[8]. However, as a result, the video features and image features lie in heterogeneous spaces, and this makes face retrieval across the image and video domains challenging.

In this paper, we propose a heterogeneous hashing network (HHN) to generate isomorphic binary hash codes of face images and videos for face retrieval across image and video domains. Our network contains an image branch and a video branch to project both face images and videos into a common space, as shown in Fig. 1.

The two branches project face images and face videos, respectively, from heterogeneous spaces into a common space to measure similarity. Each branch has two modules: the feature extractor module and non-linear mapping module. The feature extractor modules of the two branches aim to represent face images or videos via appropriate features, such as CNN features [9]–[11] for images, and C3D features [12] or NAN features [13] for videos. The non-linear mapping modules are used to transform the two heterogeneous feature spaces (image feature space and video feature space) into a common space. The similarity between a face image and a face video can be

Manuscript received January 23, 2018; revised May 22, 2018 and July 26, 2018; accepted July 29, 2018. Date of publication August 20, 2018; date of current version February 21, 2019. This work was supported by the Natural Science Foundation of China under Grant 61472038 and Grant 61375044. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Lei Zhang. *(Corresponding author: Mingtao Pei.)*

The authors are with the Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, China (e-mail: jingchenchen1996@bit.edu.cn; dongzhen@bit.edu.cn; peimt@bit.edu.cn; jiayunde@bit.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2018.2866222

measured through the distance of their corresponding features in the common space. These features are high-dimensional vectors of floating point numbers, which cannot satisfy the requirements of low computation complexity and storage costs in the retrieval task. Non-linear hash functions are learned in the common space to get hash representations of both face images and videos. With the hash representations, it only takes nearly constant time and extremely low memory cost to compute similarities for fast retrieval.

The HHN is trained under joint supervision of the Fisher loss, the Softmax loss, and the triplet ranking loss. To enhance the discriminative power of the common space, the Fisher loss and the Softmax loss are enforced to exploit the common semantic information of face images and videos. The Fisher loss uses the difference form of within-class and between-class scatter and adopts learnable mean vectors. Thus, the Fisher loss is appropriate for the mini-batch based optimization method.

The enforcement of the two loss functions ensures the discriminative power of features in the common space and improves the commonalities of these features from the image and video domains.

The triplet ranking loss is enforced on the hash representations generated by the non-linear hash functions to further reduce the gap between the image and video domains. The enforcement of the triplet ranking loss ensures effective hash representations for the cross-domain retrieval task.

The proposed method achieves excellent results for face retrieval across image and video domains on a large scale face video dataset and two challenging TV-series datasets. In addition, the heterogeneous hashing network provides a general framework for deep learning-based cross-domain hashing methods and can be easily adopted in many other cross-domain retrieval tasks.

The remainder of the paper is organized as follow. Sec. II reviews the related work including face retrieval methods, single-modality and multi-modality hashing methods. Sec. III elaborates the heterogeneous hashing network. Sec. IV presents the experiment results of the proposed method on a large-scale face video dataset and two TV-series datasets, and Sec. V concludes this paper.

II. Related Work

A. Face Retrieval

Face image retrieval has been extensively studied [14]–[21]. These works first extract discriminative features of face images such as GIST features [16], gabor-LBP histogram [14], and CNN features [20]. Then, hashing-based methods [15], [19] or sparse coding-based methods [14], [16], [21] are utilized to obtain compact face representations for retrieval.

Recently, face video retrieval has drawn significant attention due to the tremendous explosion in video data. Li *et al.* [22] 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 extended the CVC method by representing face videos as spatial pyramid covariance matrices for retrieval and significantly improved the performance [23]. Dong *et al.* [24] proposed an end-to-end deep network to learn discriminative and compact frame representations and fuse them to obtain final video representations for retrieval.

Although enormous methods on face image and video retrieval have been proposed and have exhibited excellent performance, these methods cannot be directly used in face retrieval across image and video domains.

Li *et. al.* [25] proposed a hashing method across the Euclidean space and the Riemannian manifold to measure the similarity of face images and videos for face video retrieval with image query. They achieved performances that were superior to many traditional single-modality and multi-modality methods.

We propose a heterogeneous hashing network (HHN) to generate isomorphic binary hash codes of face images and videos for face retrieval across image and video domains. To the best of our knowledge, this is the first paper on face retrieval across image and video domains.

B. Single-Modality Hashing

Hashing methods are widely used in retrieval systems owing to the increased efficiency in both speed and storage. Existing hashing methods can be roughly divided into two types: data-independent hashing and data-dependent hashing. As a representative of data-independent hashing methods, locality sensitive hashing (LSH) [26] and its variants, kernelized locality-sensitive hashing (KLSH) [27], use random projections as hash functions.

Slightly different from LSH, the shift-invariant kernel hashing (SIKH) [28] uses a shifted cosine function to compute hash codes. Despite theoretical asymptotic guarantees, LSH, KLSH and SIKH still require long hash codes to obtain satisfactory retrieval results.

In contrast, data-dependent hashing methods, namely, learning-based methods, aim to generate compact similarity preserving hash codes through exploiting the structure or supervision information of the training data. Most data-dependent methods fall into three categories: unsupervised, semi-supervised, and supervised methods. Unsupervised methods learn hash functions only by using unlabeled training data, and the representatives are spectral hashing (SH) [29], iterative quantization hashing (ITQ) [30], and multilinear hyperplane hashing [31].

Semi-supervised and supervised methods improve hash code quality by using supervision information, such as semi-supervised hashing (SSH) [32], supervised iterative quantization hashing (SITQ) [30], kernel-based supervised hashing (KSH) [33], and adaptive hashing [34].

Recently, many deep neural network based hashing methods have been proposed, such as CNN hashing [35], deep semantic ranking-based hashing [36], unsupervised deep learning compact binary representations [37], deep supervised hashing (DSH) [38], deep pairwise-supervised hashing [39], and deep video hashing [40]. Benefiting from the powerful ability of deep neural networks to describe complex non-linear mappings and

the end-to-end training manner, these methods achieve good performance on image retrieval tasks. Motivated by these works, we present the heterogeneous hashing network to learn binary hash representations of both face images and videos for retrieval.

C. Multi-Modality Hashing

The hashing methods mentioned above achieve great success in a wide range of applications, but they are not able to retrieve data for multiple modalities. Multi-modality hashing has achieved increased attention in recent years [41]–[43].

Similar to single-modality hashing methods, existing multi-modality hashing methods can be roughly categorized into two types: unsupervised and supervised. Representative unsupervised methods include correspondence autoencoder-based hashing (CAH) [44], cross view hashing (CVH) [45], and predictable dual-view hashing (PDH) [46]. The learning criterions of CAH, CVH and PDH are reconstruction error minimization, graph-based similarity preservation, and predictability maintaining, respectively. Typical supervised multi-modality hashing methods include cross-modal similar sensitive hashing (CMSSH) [47], multimodal latent binary embedding (MLBE) [48], multi-modal neural network hashing (MMNN) [49], parametric local multimodal hashing (PLMH) [50], hashing with semantic correlation maximization (SCM) [51], and semantic preserving hashing (SPH) [42]. Supervised methods are able to fully utilize the semantic information to reduce the discrepancy and the semantic gap between modalities, and they achieved better performances than unsupervised methods.

Jiang and Li [41] proposed the deep cross-modal hashing method and applied it to the cross-modal retrieval between images and text sentences. Cao *et al.* [43] proposed the deep visual-semantic hashing method to deeply explore the heterogeneous correlation structure information for cross-modal retrieval between images and text sentences. The networks in the above two methods are designed for retrieval between images and text sentences and cannot be directly used for retrieval across image and video domains. Our heterogeneous hashing network provides a general deep architecture for multi-modality hashing and achieves excellent performance on face retrieval across image and video domains.

III. Heterogeneous Hashing Network

A. Overview

The Heterogeneous Hashing Network (HHN) can generate isomorphic binary hash codes for face images and videos to accomplish face retrieval across image and video domains.

The HHN contains two branches: the image branch and video branch, to deal with face images and videos, respectively. In each branch, a feature extractor module is equipped to characterize and represent the input image or video data, and then the non-linear mapping module is introduced to project the extracted features into the discriminative common space. The two branches with the feature extractor and non-linear mapping modules are able to discover the heterogeneous correlation structure across the image and video domains and reduce the discrepancy between the two domains. By enforcing discriminative constraints on the common space, the commonalities of features from multiple domains can be improved.

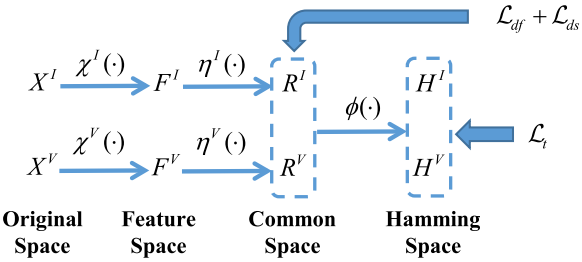


Fig. 2. Dataflow illustration of the heterogeneous hashing network.

Non-linear hash functions are learned in the common space to obtain hash representations of both face images and videos.

Overall, the HHN has five components: feature extraction module of the image branch, non-linear mapping module of the image branch, feature extraction module of the video branch, non-linear mapping module of the video branch, and the hash learning module. All the five components are implemented via neural networks, so the entire HHN actually forms a unified optimization framework for face retrieval across image and video domains. Note that the HHN can be easily extended to hashing for data in other domains in addition to image and video, such as audio and text, as long as the suitable feature extraction module is designed. Our network provides a general framework for deep learning-based cross-domain hashing methods and can be easily adopted in many other cross-domain retrieval tasks.

B. Formulation

We define that the superscript I indicates variables or functions of the face image branch, and V indicates ones of the face video branches. Fig. 2 depicts the dataflow of the HHN whose five components are represented as five functions: feature extractor of image branch $\chi^I(\cdot)$, non-linear mapping of image branch $\eta^I(\cdot)$, feature extractor of video branch $\chi^V(\cdot)$, non-linear mapping of video branch $\eta^V(\cdot)$, and the hash functions $\phi(\cdot)$. The variables in Fig. 2 include the following: $\boldsymbol{X}$ for the data in original space, $\boldsymbol{F}$ for the extracted features, $\boldsymbol{R}$ for the representations in the common space, and $\boldsymbol{H}$ for the final hash codes in the Hamming space. We thus have

$$\begin{aligned} \boldsymbol{R}^I &= \eta^I(\chi^I(\boldsymbol{X}^I)), & \boldsymbol{H}^I &= \phi(\boldsymbol{R}^I), \\ \boldsymbol{R}^V &= \eta^V(\chi^V(\boldsymbol{X}^V)), & \boldsymbol{H}^V &= \phi(\boldsymbol{R}^V). \end{aligned} \tag{1}$$

Let c be the total number of individuals, and the face image representations in the common space are denoted as $\boldsymbol{R}^I = [\boldsymbol{R}^I_1, \boldsymbol{R}^I_2, ..., \boldsymbol{R}^I_c] \in \mathbb{R}^{d\times n^I}$, where d is the dimension of the common space and n^I represents the total number of the face image samples. The face video representations in the common space are similarly defined as $\boldsymbol{R}^V = [\boldsymbol{R}^V_1, \boldsymbol{R}^V_2, ..., \boldsymbol{R}^V_c] \in \mathbb{R}^{d\times n^V}$, where n^V is the total number of face video samples. For the i-th ($i = 1, 2, ..., c$) category, we use $\boldsymbol{R}^I_i = [\boldsymbol{r}^I_{i,1}, \boldsymbol{r}^I_{i,2}, ..., \boldsymbol{r}^I_{i,n^I_i}] \in \mathbb{R}^{d\times n^I_i}$ and $\boldsymbol{R}^V_i = [\boldsymbol{r}^V_{i,1}, \boldsymbol{r}^V_{i,2}, ..., \boldsymbol{r}^V_{i,n^V_i}] \in \mathbb{R}^{d\times n^V_i}$ to describe the corresponding face image and video representations, where n^I_i and

n_i^V are the number of two types of representations of the current category, and we thus have $\sum_{i=1}^{c} n_i^K = n^K$, where $K \in \{I, V\}$

To enhance the discriminative power of the common space and simultaneously reduce the discrepancy between face image and video representations, two types of constraints are enforced to the representations in the common space, *i.e.*, the softmax loss and the Fisher loss. To guarantee the separability of the representations, which means that representations of different classes should stay apart, the softmax loss is enforced in the common space as

$$\mathcal{L}_{ds} = -\sum_{i=1}^{c} \sum_{K\in\{I,V\}} \sum_{j=1}^{n_i^K} \log \frac{e^{\boldsymbol{W}_i^\top \boldsymbol{r}_{i,j}^K + \boldsymbol{b}_i}}{\sum_{k=1}^{c} e^{\boldsymbol{W}_k^\top \boldsymbol{r}_{i,j}^K + \boldsymbol{b}_k}}. \quad (2)$$

The Fisher loss, which minimizes the intra-class variations and simultaneously maximizes the inter-class variations, is further utilized to efficiently enhance the discriminative power of the common space.

The standard Fisher loss is implemented by minimizing the Rayleigh quotient of representations of both face images and videos as $tr(\boldsymbol{S}_W)/tr(\boldsymbol{S}_B)$, where $tr(\cdot)$ represents the trace of the square matrix inside. $\boldsymbol{S}_W$ and $\boldsymbol{S}_B$ represent the within-class and between-class scatter matrices, respectively, and are given by

$$\begin{aligned} \boldsymbol{S}_W &= \sum_{i=1}^{C} \sum_{K\in\{I,V\}} \sum_{j=1}^{n_i^K} (\boldsymbol{r}_{i,j}^K - \boldsymbol{\mu}_i)(\boldsymbol{r}_{i,j}^K - \boldsymbol{\mu}_i)^\top, \\ \boldsymbol{S}_B &= \sum_{i=1}^{C} n_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^\top, \end{aligned} \quad (3)$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}$ are the mean vector of the i-th individual and all samples:

$$\boldsymbol{\mu}_i = \frac{1}{n_i} \sum_{K\in\{I,V\}} \sum_{j=1}^{n_i^K} \boldsymbol{r}_{i,j}^K, \quad \boldsymbol{\mu} = \frac{1}{n} \sum_{i=1}^{C} n_i \boldsymbol{\mu}_i, \quad (4)$$

where $n_i = n_i^I + n_i^V$ and $n = n^I + n^V = \sum_{i=1}^{c} n_i$ represent the sample number of the i-th individual and all individuals, respectively. Since $\boldsymbol{S}_W$ and $\boldsymbol{S}_B$ are computed by using face data over both the image domain and the video domain, the utilization of the Fisher loss reduces the discrepancy between the two domains and simultaneously takes the discriminative power of representations into account.

To effectively minimize the Rayleigh quotient, both $\boldsymbol{S}_W$ and $\boldsymbol{S}_B$ should be non-singular matrices. Thus, the dimension of the representations must be smaller than the number of categories. This requirement cannot be satisfied in many real applications in which representations for hash learning usually have high dimensions. In addition, since the mean vectors are computed over the whole training set, the Fisher loss in the Rayleigh quotient form cannot be optimized by the mini-batch based optimization method, such as the stochastic gradient descend method, which is widely used in training deep neural networks.

We introduce the difference form of Fisher loss as $tr(\boldsymbol{S}_W) - tr(\boldsymbol{S}_B) + \lambda\|\boldsymbol{R}\|_F$ to overcome the problems mentioned above, where $\|\boldsymbol{R}\|_F$ is added to ensure the convexity of the objective function [52], $\|\cdot\|_F$ is the Frobenius norm of the inside matrix, and λ is a trade-off hyper-parameter. Furthermore, we keep the mean vectors learnable for the feasibility of the mini-batch optimization methods. In the training process, we simultaneously update the mean vectors, minimize the distances between the features and their corresponding class mean vectors, and maximize the distances between each class mean vector and the mean vector of all the features.

Intuitively, the Fisher loss simultaneously pulls the features of the same class to their mean vectors and pushes the mean vectors of each class away from the mean vector of all features. The Fisher loss is thus equally formulated as

$$\begin{aligned} \mathcal{L}_{df} &= \lambda\|\boldsymbol{R}\|_F^2 \\ &+ \frac{1}{2n} \sum_{i=1}^{c} \sum_{\substack{K\in \\ \{I,V\}}} \sum_{j=1}^{n_i^K} \|\boldsymbol{r}_{i,j}^K - \boldsymbol{\mu}_i\|_2^2 - \frac{1}{2n} \sum_{i=1}^{c} n_i \|\boldsymbol{\mu}_i - \boldsymbol{\mu}\|_2^2. \end{aligned} \quad (5)$$

In the common space, non-linear hash functions are learned. To obtain isomorphic hash representations for retrieval, we encourage the hash functions to hold a large margin between the distances of positive and negative pairs of hash representations. To this end, the triplet ranking loss is exploited in the Hamming space. The triplet ranking loss indicates that the relative similarities of the hash representations in a form such as "face $\boldsymbol{h}$ is more similar to $\widetilde{\boldsymbol{h}}$ than $\widehat{\boldsymbol{h}}$", where $\boldsymbol{h}$, $\widetilde{\boldsymbol{h}}$, and $\widehat{\boldsymbol{h}}$ are the anchor point, the positive sample and the negative sample, respectively. Let $\boldsymbol{h}$ and $\widetilde{\boldsymbol{h}}$ be samples from the same individual, while $\widehat{\boldsymbol{h}}$ belongs to a different individual, $(\boldsymbol{h}, \widetilde{\boldsymbol{h}})$ forms a positive sample pair and $(\boldsymbol{h}, \widehat{\boldsymbol{h}})$ is the negative sample pair. Thus, the loss of one triplet is formulated as

$$t(\boldsymbol{h}, \widetilde{\boldsymbol{h}}, \widehat{\boldsymbol{h}}) = \max\Big(d\big(\boldsymbol{h}, \widetilde{\boldsymbol{h}}\big) - d\big(\boldsymbol{h}, \widehat{\boldsymbol{h}}\big) + \zeta, 0\Big), \quad (6)$$

where $d(\boldsymbol{\theta_1}, \boldsymbol{\theta_2}) = (l - \boldsymbol{\theta}_1^\top \boldsymbol{\theta_2})/2$ is the Hamming distance in the binary space, l is the bit number of hash codes, and $\zeta \geq 0$ represents the margin of the distance differences between positive and negative pairs.

Aiming to eliminate the discrepancy between the two domains and ensure good optimization, it is crucial to generate effective triplets for triplet ranking loss because many triplets whose samples belong to the same domain are invalid for the cross-domain retrieval task. In addition, many triplets' losses are approximately zero, which would result in slower convergence. Thus, we simplify the problem of generating triplets down to the problem of selecting the negative sample $\widehat{\boldsymbol{h}}$ for the positive sample pair $(\boldsymbol{h}, \widetilde{\boldsymbol{h}})$ from the whole batch. For each positive sample pair, we set $\boldsymbol{h}$ and $\widetilde{\boldsymbol{h}}$ to belong to the same individual but different domains to generate triplets that are beneficial to our task.

Specifically, we organize the training set in the form of positive sample pairs to let a batch with L samples have $L/2$ positive sample pairs. Each positive sample pair contains a face image and a face video. For a positive sample pair $(\boldsymbol{h}, \widetilde{\boldsymbol{h}})$, we select negative samples only from the left $L-2$ samples of the batch.

Two conditions must be met so that the negative sample $\widehat{\boldsymbol{h}}$: $\widehat{\boldsymbol{h}}$ belongs to a different individual from $\boldsymbol{h}$ and $\widetilde{\boldsymbol{h}}$, and the triplet $(\boldsymbol{h}, \widetilde{\boldsymbol{h}}, \widehat{\boldsymbol{h}})$ has positive loss, *i.e.* $d(\boldsymbol{h}, \widetilde{\boldsymbol{h}}) - d(\boldsymbol{h}, \widehat{\boldsymbol{h}}) + \zeta > 0$. Let $\mathbb{N}$ be the negative sample set for pair $(\boldsymbol{h}, \widetilde{\boldsymbol{h}})$; we select M negative samples from $\mathbb{N}$ in two ways: hard negative selecting and random negative selecting.

- *Hard Negative Selecting:* Hard negative samples mean that they are much closer to $\boldsymbol{h}$ than other negative samples. Let $\mathbb{Q}$ be the set of hard negative samples; we have

$$\max_{\boldsymbol{q} \in \mathbb{Q}} d(\boldsymbol{h}, \boldsymbol{q}) < \min_{\boldsymbol{q} \in \mathbb{N} - \mathbb{Q}} d(\boldsymbol{h}, \boldsymbol{q}). \tag{7}$$

  We enforce that $|\mathbb{Q}| = M_1$.
- *Random Negative Selecting:* As for other M_2 negative samples where $M_2 = N - M_1$, we randomly select them from $\mathbb{N} - \mathbb{S}$.

The percentage of hard negative samples, $\eta = M_1/M$, is set as 0.5 in our experiments. For each positive sample pair, We first set the face image as the anchor point and the face video as the positive points and select M negative samples from the batch to form M triplets, which forms the $\mathbb{T}_1$. Then, we set the face video as the anchor points and M triplets is generated in the same way and formed $\mathbb{T}_2$.

Note that we shuffle the training set at the beginning of each epoch to generate as many appropriate triplets as possible.

By adding the domain constraint to the positive sample pair and exploiting the hard negative selecting and random negative selecting strategy, effective triplets are obtained, and the triplet ranking loss is finally given by

$$\mathcal{L}_t = \sum_{(\boldsymbol{h_1^I}, \boldsymbol{h_2^V}) \in \mathbb{P}} \left(\sum_{\boldsymbol{h_3} \in \mathbb{T}_1} t(\boldsymbol{h_1^I}, \boldsymbol{h_2^V}, \boldsymbol{h_3}) + \sum_{\boldsymbol{h_3} \in \mathbb{T}_2} t(\boldsymbol{h_2^V}, \boldsymbol{h_1^I}, \boldsymbol{h_3}) \right) \tag{8}$$

where $\mathbb{P}$ is the set of positive sample pairs in a mini-batch.

Considering the discriminative constraint of the common space and the triplet ranking loss of the binary space simultaneously, our HHN is formulated as

$$\min_{\boldsymbol{\Gamma}, \boldsymbol{\Theta}, \boldsymbol{\Lambda}} \quad \mathcal{L} = \alpha \mathcal{L}_{ds} + \beta \mathcal{L}_{df} + \mathcal{L}_t, \tag{9}$$

where $\boldsymbol{\Gamma} = \{\chi^I(\cdot), \eta^I(\cdot), \chi^V(\cdot), \eta^V(\cdot), \phi(\cdot)\}$ represent all the parameters in the HHN, $\boldsymbol{\Theta} = \{\boldsymbol{W}_i, \boldsymbol{b}_i | i = 1, 2, ..., c\}$ is the parameter of the softmax loss $\mathcal{L}_{ds}$, $\boldsymbol{\Lambda} = \{\boldsymbol{\mu}_i | i = 1, 2, ..., c\}$ is the parameter of the Fisher loss $\mathcal{L}_{df}$, and α and β are hyper-parameters to balance the importance between terms.

C. Optimization

We use the classical back-propagation method to train the HHN modeled by Eq.(9), and the back-propagation is implemented via the stochastic gradient descend optimization method. The gradients of the losses $\mathcal{L}_{ds}$, $\mathcal{L}_{df}$ and $\mathcal{L}_t$ are necessary for the optimization. The $\mathcal{L}_{ds}$ is the well-known softmax loss whose gradients w.r.t. $\boldsymbol{\Theta}$ and $\boldsymbol{R}$ can be easily calculated.

The gradients of $\mathcal{L}_{df}$ w.r.t. $\boldsymbol{\mu}_i$ and $\boldsymbol{r}_{i,j}^K (K \in \{I, V\})$ are given by

$$\frac{\partial \mathcal{L}_{df}}{\partial \boldsymbol{\mu}_i} = \frac{1}{n} \sum_{K \in \{I,V\}} \sum_{j=1}^{n_i^K} (\boldsymbol{\mu}_i - \boldsymbol{r}_{i,j}^K) - \frac{n_i}{n} (\boldsymbol{\mu}_i - \boldsymbol{\mu}),$$

$$\frac{\partial \mathcal{L}_{df}}{\partial \boldsymbol{r}_{i,j}^K} = \frac{1}{n} (\boldsymbol{r}_{i,j}^K - \boldsymbol{\mu}_i) + 2\boldsymbol{r}_{i,j}^K. \tag{10}$$

For triplet ranking loss in Eq. (8), we list the gradients of Eq.(6) w.r.t. $(\boldsymbol{h}, \widetilde{\boldsymbol{h}}, \widehat{\boldsymbol{h}})$ as

$$\frac{\partial t}{\partial \boldsymbol{h}} = \frac{1}{2} (\widehat{\boldsymbol{h}} - \widetilde{\boldsymbol{h}}) \times \mathbf{1}(\Lambda),$$

$$\frac{\partial t}{\partial \widetilde{\boldsymbol{h}}} = -\frac{1}{2} \boldsymbol{h} \times \mathbf{1}(\Lambda), \quad \frac{\partial t}{\partial \widehat{\boldsymbol{h}}} = \frac{1}{2} \boldsymbol{h} \times \mathbf{1}(\Lambda),$$

$$\Lambda \triangleq d(\boldsymbol{h}, \widetilde{\boldsymbol{h}}) - d(\boldsymbol{h}, \widehat{\boldsymbol{h}}) + \zeta > 0, \tag{11}$$

where $\mathbf{1}(\cdot)$ is the indicator function that returns 1 if the condition inside is true and 0 for other occasions.

Since the $\mathcal{L}_{ds}$ and $\mathcal{L}_{df}$ are enforced on the representations in the common space and the $\mathcal{L}_t$ is conducted on the hash representations as depicted in Fig. 2, we organize the optimization method in two procedures: pre-training and fine-tuning. In the pre-training procedure, we optimize the image and video branches except for the hash functions to obtain good initializations for network optimization. The pre-training procedure provides effective feature extractors and non-linear mappings that project face images and videos from heterogeneous spaces to vectors in a common space. Since only $\chi^I(\cdot), \eta^I(\cdot), \chi^V(\cdot)$ and $\eta^V(\cdot)$ are learned while keeping the hash functions $\phi(\cdot)$ fixed during the optimization in this procedure, only $\mathcal{L}_{ds}$ and $\mathcal{L}_{df}$ are used. Benefiting from the softmax and Fisher losses, the generated representations of both face images and videos are individual separable and discriminative, which thus significantly reduces the discrepancy between domains. The fine-tuning procedure optimizes the entire network with the initializations obtained in the pre-training procedure. This procedure aims to integrate the image branch, the video branch, and the hash functions into a unified optimization framework so that all the components interact with each other for the final retrieval task. The robustness of image and video branches influence the hashing performance, and the hashing performance inversely guides the learning of the two branches.

Overall, the optimization of the HHN is summarized in Algorithm 1, where $\boldsymbol{\Phi} = \{\chi^I(\cdot), \eta^I(\cdot), \chi^V(\cdot), \eta^V(\cdot)\}$ represents the parameters of the image and video branches and $\boldsymbol{\Gamma} = \boldsymbol{\Phi} \cup \{\phi(\cdot)\}$ represents the parameters of the whole network.

D. Implementation Details

In our implementation, the inputs of the HHN are face images and kernel matrices representing face videos. For a face video, we execute PCA on its frames to get 100-dim vectors and then calculate a 100×100 RBF kernel matrix $\boldsymbol{K}$ by using the method in [8]. In our network, the image branch consists of stacked fully connected layers and ReLU activation layers, and the size of the image branch is "$h \times w - 100 - 512 - 1024 - 100$", where h

Algorithm 1: Heterogeneous Hashing Network Training Algorithm

Input: Face Images $\boldsymbol{X}^I$, Face Videos $\boldsymbol{X}^V$, their corresponding labels, and hyper-parameters α and β.
Output: HHN Weights $\boldsymbol{\Gamma}$.

1. Pre-Training
$t \leftarrow 0$;
Randomly initialize $\boldsymbol{\Psi}^t = \boldsymbol{\Phi}^t \cup \boldsymbol{\Theta}^t \cup \boldsymbol{\Lambda}^t$;
repeat
  Forward: Calculate the joint losses of softmax and Fisher $\mathcal{L} = \alpha\mathcal{L}_{ds} + \beta\mathcal{L}_{df}$;
  Backward: Calculate gradients to the input layer by layer, and gradients to weights $\partial\mathcal{L}/\partial\boldsymbol{\Psi}^t$;
  Update: $\boldsymbol{\Psi}^t = \boldsymbol{\Psi}^{t+1} - \gamma^t(\partial\mathcal{L}/\partial\boldsymbol{\Psi}^t)$;
  $t \leftarrow t + 1$;
until Convergence;
2. Fine-Tuning
$s \leftarrow 0$;
Initialize $\boldsymbol{\Psi}$ with the pre-trained results and initialize the hash functions $\phi(\cdot)$ randomly: $\boldsymbol{\Omega}^s = \boldsymbol{\Psi}^t \cup \{\phi^s(\cdot)\}$;
repeat
  Forward: Calculate the losses $\mathcal{L} = \mathcal{L}_t$;
  Backward: Calculate gradients to the input layer by layer, and gradients to weights $\partial\mathcal{L}/\partial\boldsymbol{\Omega}^s$;
  Update: $\boldsymbol{\Omega}^{s+1} = \boldsymbol{\Omega}^s - \gamma^s(\partial\mathcal{L}/\partial\boldsymbol{\Omega}^s)$;
  $s \leftarrow s + 1$;
until Convergence;
Return $\boldsymbol{\Gamma}$.

and w represent the height and width of the face images and the other numbers represent the neuron number of the fully connected layers. In the video branch, a vectorization operation layer is inserted and followed by the stacked fully connected layers and ReLU activation layers, and the size of the video branch is "$100 \times 100 - 5050 - 100 - 512 - 1024 - 100$". According to [53], the vectorization operation layer vectorizes the kernel matrix in the form of

$$vec(\boldsymbol{K}) = [\boldsymbol{V}_{1,1}, \sqrt{2}\boldsymbol{V}_{1,2}, ..., \sqrt{2}\boldsymbol{V}_{1,p}, \boldsymbol{V}_{2,2}, \sqrt{2}\boldsymbol{V}_{2,3}, ..., \sqrt{2}\boldsymbol{V}_{2,p}, ..., \sqrt{2}\boldsymbol{V}_{p-1,p}, \boldsymbol{V}_{p,p}]^\top, \tag{12}$$

to produce a $p(p+1)/2$-dim vector, where $p = 100$, and $\boldsymbol{V} = \log(\boldsymbol{K})$ is the matrix logarithm of $\boldsymbol{K}$. In each branch, the feature extractor module is exploited to represent the input image or video as a 100-dim vector, and the following non-linear mapping module projects features in heterogeneous spaces into the common space. The hashing functions are implemented by a "$100 - 100 - l$" network with two fully connected layers, where l is the bit number of the hash codes.

After the last fully connected layer of the hashing functions, a $\tanh$ layer is inserted to approximate the $sgn(\cdot)$ function for the quantization of the hash codes. Furthermore, the hyper-parameters, α, β and λ, are set as 1, 0.1 and 0.001, respectively.

Both of the pre-training and fine-tuning procedures are implemented by using the open source Caffe tool [54]. The pre-training procedure is optimized via the stochastic gradient descent method, where the momentum and the weight decay are set as 0.9 and 5×10^{-4}, respectively. The learning rate of the optimization is initialized as 0.01 and decreased according to the polynomial policy with a power value of 0.8. The size of the mini-batch of the training samples is set as 512, and the total number of the iterations is 100 K. The time cost of each iteration of the pre-training procedure is about 27 ms using an NVIDIA Titan X GPU. The total training time of the pre-training procedure is about 50 min. The memory cost of the pre-training procedure is about 350 MB.

Similarly, the momentum and the weight decay of the fine-tuning procedure are 0.8 and 5×10^{-5}, respectively. The learning rate of the optimization is initialized as 0.001 and decreased according to the polynomial policy with a power value of 0.8. The size of the mini-batch of the training samples is set as 512, and the total number of the iterations is 50 K. The time cost of each iteration and the total training time of the fine-tuning procedure are about 25 ms and 30 min, respectively. The memory cost of the fine-tuning procedure is about 350 MB.

IV. Experiments

A. Datasets and Experimental Setting

To evaluate the performance of the proposed method, we conduct experiments on two datasets: ICT-TV and Celebrity-1000. The ICT-TV dataset [55] has two large-scale face video shot collections from the first seasons of two popular American shows: the Big Bang Theory (BBT) and Prison Break (PB). The filming styles of the two TV-series are quite different. The BBT is an indoor melodrama with only 5 main characters, and each episode lasts about 20 minutes. In contrast, the PB mostly takes place outside, and the average length of all the episodes is around 42 minutes, which leads to large illumination variations. The total number of face video shots of the two collections are 4,667 and 9,435, respectively.

Different from ICT-TV, Celebrity-1000 (Celeb1K) [56] is a large-scale face video dataset. It contains 159,726 video sequences of 1,000 human subjects, with 2.4M frames in total (about 15 frames per sequence). The face frames of videos in the dataset have been preprocessed by detection, alignment, and resized to 64×48.

Since the two datasets are not specially proposed for the cross-domain face retrieval task and only contain face videos, we use two approaches to form the image sets. A common method, which has been adopted by recent work on video retrieval with image query task [25], [57], [58], is to select frames from the videos as the image set. Compared with frames selected from videos, images from the Internet contain more variations and are thus more challenging [59]. To fully evaluate the performance of the proposed method, we conduct experiments under both settings, according to the two sources of the images, *i.e.*, images selected from the videos and images from the Internet.

On all of the BBT, the PB and the Celeb1K datasets, subsets are selected as the training sets. Specifically, all the individuals in the datasets are first sorted according to the number of videos per individual from large to small, and the top 5 of the BBT,

TABLE I
COMPARISON MAPS OF QUERY-BY-IMAGE VIDEO RETRIEVAL ON THE TWO DATASETS USING SELECTED FRAMES

Type	Methods	*the Big Bang Theory*				*Prison Break*				*Celebrity 1000*			
		8 bits	16 bits	32 bits	64 bits	8 bits	16 bits	32 bits	64 bits	8 bits	16 bits	32 bits	64 bits
single modality	LSH [26]	0.2448	0.2919	0.3162	0.3726	0.1764	0.1728	0.1868	0.1857	0.0912	0.1130	0.1545	0.1915
	RR [30]	0.2916	0.3551	0.3571	0.3929	0.1689	0.1786	0.1872	0.1896	0.1156	0.1447	0.1904	0.2094
	ITQ [30]	0.3546	0.3599	0.3898	0.3998	0.1746	0.1728	0.1794	0.1860	0.1169	0.1686	0.2054	0.2317
	SH [29]	0.2881	0.3421	0.3145	0.3177	0.1685	0.1752	0.1903	0.2028	0.1276	0.1648	0.1935	0.2063
	SSH [32]	0.3432	0.2761	0.2541	0.2447	0.1838	0.1840	0.1799	0.1823	0.1342	0.1765	0.1823	0.1568
	KSH [33]	0.7825	0.7988	0.8638	0.8262	0.3420	0.4006	0.4260	0.4414	0.2796	0.3468	0.3841	0.4383
	SITQ [30]	0.4703	0.3818	0.3837	0.3379	0.1893	0.1818	0.1824	0.1831	0.1541	0.1937	0.2280	0.2341
	DSH [38]	0.8820	0.8926	0.9179	0.9356	0.6206	0.6811	0.7406	0.7992	0.7023	0.7585	0.7950	0.8085
multiple modality	CCA [30]	0.3664	0.3067	0.2763	0.2440	0.2193	0.2363	0.2198	0.1982	0.1646	0.1950	0.1814	0.1470
	CMSSH [47]	0.5582	0.5985	0.5794	0.5489	0.2567	0.2397	0.2162	0.2078	0.1738	0.1652	0.1500	0.1402
	PDH [46]	0.2252	0.2223	0.2817	0.2669	0.2316	0.2318	0.2131	0.2142	0.1256	0.1264	0.1191	0.1229
	MMNN [49]	0.6322	0.7296	0.7860	0.7819	0.4076	0.5522	0.5328	0.7126	0.1735	0.3656	0.3918	0.3672
	SCM [51]	0.7659	0.8461	0.8690	0.8605	0.4946	0.6165	0.6247	0.6927	0.2654	0.3370	0.4157	0.4809
	HER [25]	0.8017	0.8121	0.8677	0.8673	0.5009	0.5586	0.6348	0.6879	0.7324	0.7740	0.7659	0.7970
Our HHN		**0.9393**	**0.9448**	**0.9481**	**0.9592**	**0.7419**	**0.7788**	**0.8022**	**0.8231**	**0.7473**	**0.7585**	**0.8019**	**0.8270**

TABLE II
COMPARISON MAPS OF QUERY-BY-VIDEO IMAGE RETRIEVAL ON THE TWO DATASETS USING SELECTED FRAMES

Type	Methods	*the Big Bang Theory*				*Prison Break*				*Celebrity 1000*			
		8 bits	16 bits	32 bits	64 bits	8 bits	16 bits	32 bits	64 bits	8 bits	16 bits	32 bits	64 bits
single modality	LSH [26]	0.2300	0.2969	0.3410	0.3426	0.1641	0.1817	0.1899	0.1850	0.0768	0.1096	0.1486	0.1946
	RR [30]	0.3013	0.2969	0.3410	0.3426	0.1776	0.1728	0.1876	0.1934	0.0989	0.1559	0.1813	0.2160
	ITQ [30]	0.3213	0.3524	0.3779	0.3986	0.1824	0.1745	0.1779	0.1869	0.1130	0.1632	0.2028	0.2302
	SH [29]	0.2728	0.3336	0.3120	0.3098	0.1699	0.1762	0.1898	0.2014	0.1193	0.1645	0.1965	0.2093
	SSH [32]	0.3266	0.2809	0.2571	0.2510	0.1900	0.1859	0.1832	0.1853	0.1196	0.1761	0.1897	0.1638
	KSH [33]	0.7296	0.7852	0.8459	0.8023	0.3337	0.3920	0.4246	0.4309	0.2496	0.3366	0.3773	0.4484
	SITQ [30]	0.4451	0.3787	0.3895	0.3600	0.1951	0.1847	0.1810	0.1847	0.1270	0.1827	0.2225	0.2318
	DSH [38]	0.8707	0.8850	0.9085	0.9362	0.6160	0.6739	0.7302	0.7711	0.7225	0.7555	0.7707	0.8031
multiple modality	CCA [30]	0.3688	0.3057	0.2739	0.2423	0.2184	0.2375	0.2203	0.1977	0.1634	0.1917	0.1801	0.1458
	CMSSH [47]	0.5905	0.6109	0.6096	0.5834	0.2084	0.1946	0.1817	0.1693	0.1725	0.1601	0.1395	0.1269
	PDH [46]	0.2298	0.2278	0.2271	0.2517	0.2290	0.2292	0.2083	0.2076	0.1193	0.1213	0.1150	0.1159
	MMNN [49]	0.6694	0.7552	0.8193	0.8256	0.3980	0.5522	0.5500	0.7067	0.1851	0.3818	0.4018	0.3878
	SCM [51]	0.8079	0.8814	0.8966	0.8907	0.5195	0.6400	0.6458	0.7110	0.2666	0.3351	0.4094	0.4733
	HER [25]	0.6567	0.7214	0.7354	0.7888	0.4990	0.5773	0.6580	0.6947	0.5909	0.7260	0.7658	0.8177
Our HHN		**0.9472**	**0.9564**	**0.9537**	**0.9602**	**0.7528**	**0.7814**	**0.8143**	**0.8333**	**0.7612**	**0.7786**	**0.7974**	**0.8323**

the top 7 of the PB (except for the "Unknown" category), and top 15 of Cele1K are then selected. Each selected individual randomly provides 200 videos to form the training video set. On the first setting, one frame is randomly selected from each video to form the set of training images. And the size of the training sets are thus as follows: 1000 images and 1000 videos for BBT, 1400 images and 1400 videos for PB, and 3000 images and 3000 videos for Cele1K. The remaining videos are used as the testing video set, and the testing image set also comprises randomly selected frames from the videos, like the training image set. On the second setting, we first crawled images from typical image search engines, *i.e.*, Google, Baidu, and Bing, for selected individuals of the BBT and the PB. We did not extend the Celebrity-1000 dataset since the name list of this dataset is unavailable. Then, the face detector [60] is utilized to exclude the images that contain no faces. We further manually sifted the crawled images to ensure that the face images belong to the corresponding individual. The average number of images per individual in the BBT and the PB is about 400. For each dataset, we randomly select 200 images for each individual to form the set of training images.

The remaining images and the remaining videos formed the testing set. For both settings, 10 face images or video shots are randomly selected to form the query set, and the database consists of the remaining face images or video shots in the testing phase.

To fully evaluate the performance of our approach, we compared it with eight single-modality and six multi-modality hashing methods.

1) Single-modality hashing methods: LSH [26], SH [29], ITQ [30], SITQ [30], RR [30], SSH [32], KSH [33] and DSH[38];
2) Multi-modality hashing methods: CCA [30], PDH [46], CMSSH [47], MMNN [49], SCM [51], HER [25].

The bit number of the hash codes ranges from 8 to 64 to show the performances of all these methods versus code lengths. The parameters of the comparison methods are carefully set based on the suggestions in their original publications for fair comparison.

Four evaluation criterion are used: Precision Recall curve (PR curve), Precision curve w.r.t. Number of top returned samples (PN curve), Recall curve w.r.t. Number of top returned samples (RN curve) and mean Average Precision (mAP). The reported results including mAPs and curves are the averages of 30 rounds of tests. Only the PR, PN and RN curves under the first setting with a hash code length of 64 are presented as representative results in the interest of space.

B. Results on Selected Frames

This part of the experiments is divided into two parts, corresponding to the two types of cross-domain retrieval tasks (*i.e.*, query-by-image video retrieval and query-by-video image retrieval). For query-by-image video retrieval, the query set contains 10 face images of each person, and the database consists of the remaining video shots. To the contrary, for query-by-video image retrieval, the query set contains 10 video shots of each person, and the database consists of the remaining face images. Table I and Table II list the mAPs of all the methods

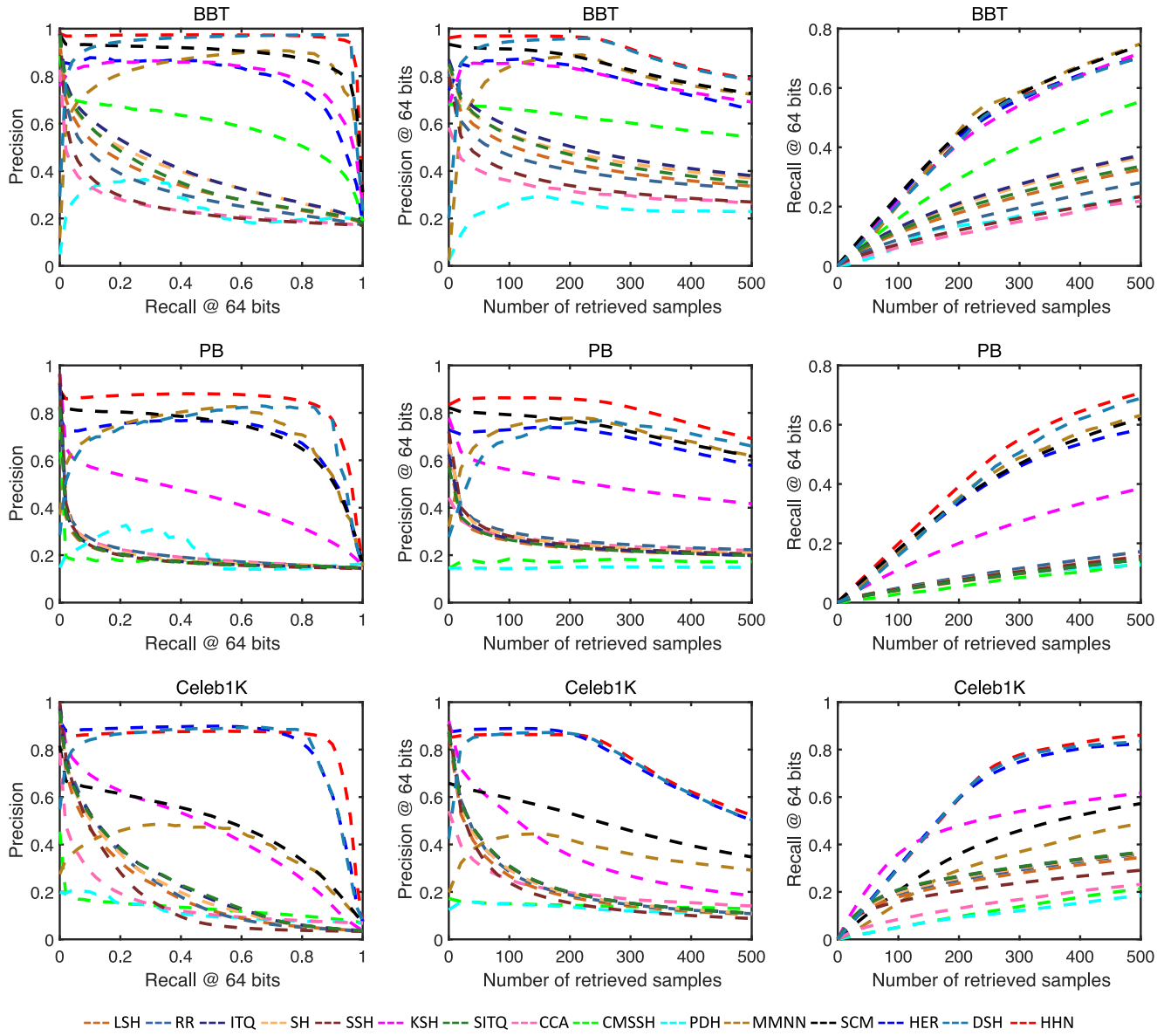


Fig. 3. Comparisons of PR, PN, and RN curves of query-by-image video retrieval on the ICT-TV and Celeb1K datasets using selected frames.

of the query-by-image video retrieval and query-by-video image retrieval, respectively. And Fig. 3 and Fig. 4 depict the corresponding comparisons of the PR, PN, and RN curves. For fair comparison, the inputs of the other methods are the features extracted by the feature extractor modules of our HHN. To use single-modality hashing methods for cross-domain face retrieval, the face video is regarded as a set of face frames, and the distance between a face image and a face video is computed by averaging the distances between the image and all the frames of the video. For multi-modality hashing methods, we first vectorize the kernel matrix in the form of Eq. (12) and then conduct hashing methods on the image feature vectors and vectorized kernel matrix since these multi-modality hashing methods except HER can only deal with the situation where modalities are represented in Euclidean space.

From the tables and figures, we find that the multi-modality hashing methods outperform single-modality hashing methods in general, and this is partially because the multi-modality methods use the kernel matrix video representation, which makes full use of correlation information between frames to characterize the complex variations in face videos. Benefiting from the powerful ability for describing complex non-linear mappings, deep hashing methods (DSH [38] and MMNN [49]) and kernel-based hashing methods (KSH [33] and HER [25]) outperform the others markedly. Supervised hashing methods achieve better performance than unsupervised methods in most cases whether in the single-modality or multi-modality hashing method; supervised methods take full advantage of the label information during hash function learning. In addition, since the videos of the Celebrity-1000 are relatively low quality and exhibit huge diversity in lighting and background, the Celebrity-1000 is more challenging than the ICT-TV dataset. As a result, all the methods perform worse in the Celebrity-1000.

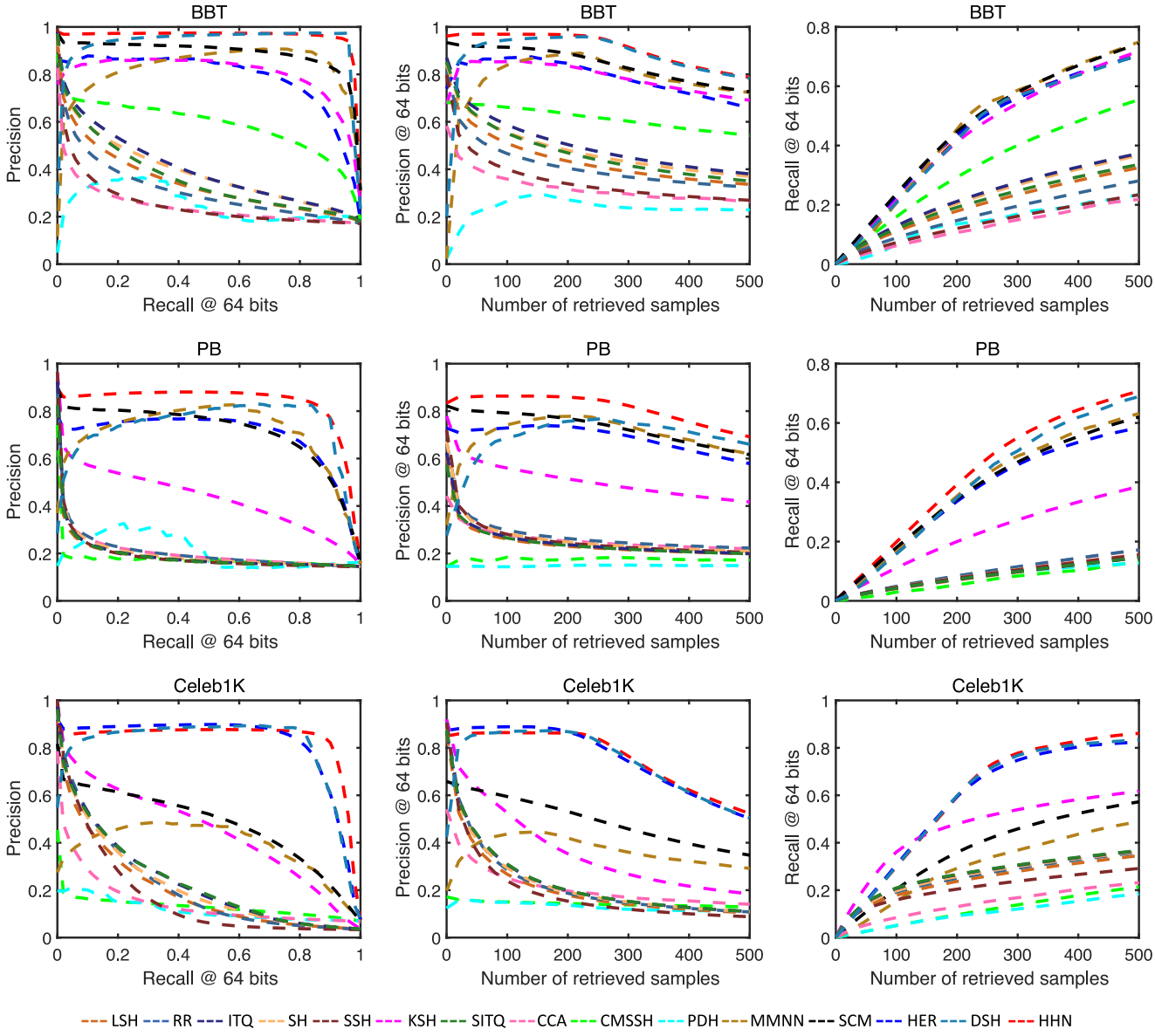


Fig. 4. Comparisons of PR, PN, and RN curves of query-by-video image retrieval on the ICT-TV and Celeb1K datasets using selected frames.

Table I and Fig. 3 show that our HHN outperforms all the other hashing methods significantly in most cases. This is true for the following reasons: (1) Both the inter-domain and intra-domain discriminativity are considered to optimize the network; (2) Both of the representations in the common space and the final hash codes are enforced to be discriminative with three losses; (3) The two-stage optimization method first projects data from heterogeneous spaces into a common space and then concentrates on learning hash functions in the discriminative space; (4) The network does not enforce any prior assumption on the data and the mapping from original spaces to the binary space so that the optimal network only depends on the data, while the kernel functions in the HER method might not be suitable all the time.

C. Results on Internet Images

Experiments on Internet images are also divided into query-by-image video retrieval and query-by-video image retrieval. Table IV and Table V list the mAPs of all the methods of the query-by-image video retrieval and query-by-video image retrieval, respectively. Since single-modality hashing methods cannot simultaneously handle both the frames in the videos and the images crawled from the Internet, only the multi-modality methods are compared with our method. For fair comparison, the inputs of the multi-modality methods are the features extracted by the feature extractor modules of our HHN.

From the tables, we find that since the crawled images contain more variations, the performances of all the methods decreased compared with the results in Sec. IV-B. However, thanks to the two-branch network architecture, the two-stage optimization strategy, and the joint supervision of the three losses, our HHN outperforms all the other multi-modality hashing methods significantly.

Figure 5 shows a real retrieval case on the PB dataset with 64-bit binary codes. It is clearly shown in the figure that the videos in the PB dataset has large variations in pose, background,

TABLE III
COMPARISON MAPS OF OUR METHODS OF QUERY-BY-IMAGE VIDEO RETRIEVAL ON THE TWO DATASETS USING SELECTED FRAMES

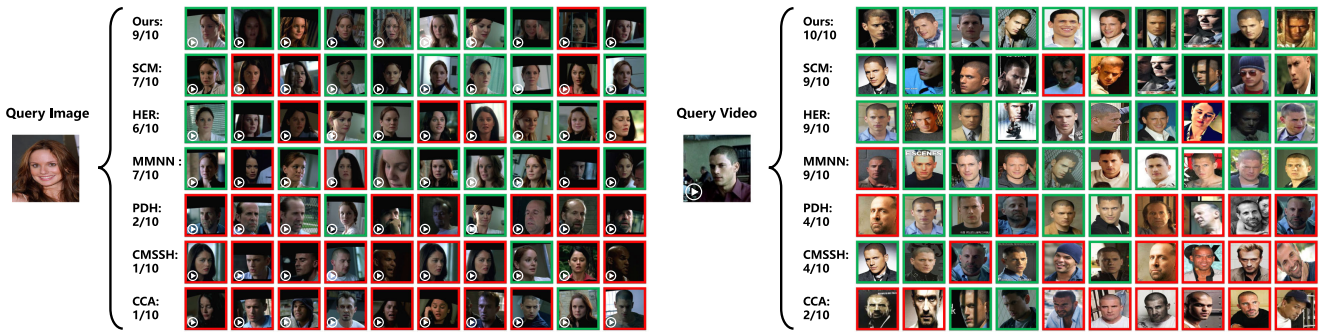
Methods	*the Big Bang Theory*				*Prison Break*				*Celebrity 1000*			
	8 bits	16 bits	32 bits	64 bits	8 bits	16 bits	32 bits	64 bits	8 bits	16 bits	32 bits	64 bits
Our HHN (S-T)	0.8095	0.8446	0.8947	0.9058	0.7049	0.7430	0.7779	0.7991	0.7324	0.7433	0.7739	0.7945
Our HHN (S-F)	0.8738	0.8977	0.9140	0.9291	0.7073	0.7499	0.7632	0.7821	0.6751	0.7513	0.7716	0.8090
Our HHN (S-F-T(r))	0.8579	0.8751	0.9078	0.9160	0.6893	0.7624	0.7812	0.8053	0.6841	0.7639	0.7878	0.7963
Our HHN (S-F-T)	**0.8807**	**0.9295**	**0.9337**	**0.9444**	**0.7419**	**0.7788**	**0.8022**	**0.8231**	**0.7473**	**0.7585**	**0.8019**	**0.8270**

TABLE IV
COMPARISON MAPS OF QUERY-BY-IMAGE VIDEO RETRIEVAL ON THE TWO DATASETS USING INTERNET IMAGES

Type	Methods	*the Big Bang Theory*				*Prison Break*			
		8 bits	16 bits	32 bits	64 bits	8 bits	16 bits	32 bits	64 bits
multiple modality	CCA [30]	0.3552	0.3112	0.2629	0.2331	0.3616	0.2686	0.2196	0.1911
	CMSSH [47]	0.4542	0.4750	0.4670	0.4529	0.2288	0.2108	0.1990	0.1854
	PDH [46]	0.3142	0.3153	0.3177	0.3212	0.2854	0.2854	0.2857	0.2785
	MMNN [49]	0.6619	0.7592	0.7503	0.7710	0.4040	0.4404	0.4969	0.5093
	SCM [51]	0.6701	0.7363	0.7658	0.7848	0.5168	0.5826	0.6238	0.6648
	HER [25]	0.7681	0.7647	0.7706	0.8061	0.4588	0.4956	0.5751	0.6463
Our HHN		**0.8609**	**0.8879**	**0.8934**	**0.9166**	**0.7029**	**0.7201**	**0.7567**	**0.7736**

TABLE V
COMPARISON MAPS OF QUERY-BY-VIDEO IMAGE RETRIEVAL ON THE TWO DATASETS USING INTERNET IMAGES

Type	Methods	*the Big Bang Theory*				*Prison Break*			
		8 bits	16 bits	32 bits	64 bits	8 bits	16 bits	32 bits	64 bits
multiple modality	CCA [30]	0.3831	0.3580	0.3029	0.2689	0.3552	0.2691	0.2223	0.1944
	CMSSH [47]	0.5230	0.5307	0.5069	0.4811	0.2041	0.1889	0.1834	0.1807
	PDH [46]	0.4499	0.4510	0.4513	0.4569	0.2713	0.2712	0.2709	0.2582
	MMNN [49]	0.7010	0.8228	0.8113	0.8305	0.4065	0.4297	0.4840	0.4875
	SCM [51]	0.7820	0.8510	0.8798	0.8930	0.4932	0.5629	0.6130	0.6562
	HER [25]	0.7517	0.7667	0.7870	0.8157	0.4857	0.5315	0.6340	0.6620
Our HHN		**0.8577**	**0.8723**	**0.9133**	**0.9315**	**0.6575**	**0.6875**	**0.6838**	**0.7440**



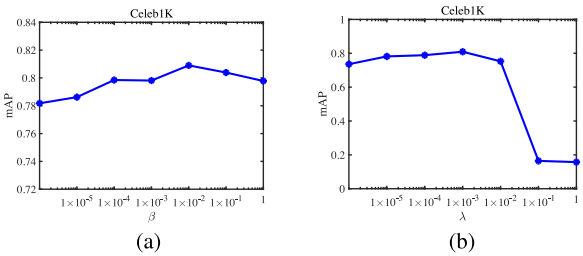
(a) (b)

Fig. 5. Real retrieval cases on the PB dataset with 64-bit binary codes. (a) Shows a query-by-image video retrieval case where the query is an image of Sara Tancredi. (b)Shows a query-by-video image retrieval case where the query is a video of Michael Scofield. Only the top-10 feedbacks of each method are shown. Red rectangles highlight the errors.

expression and illumination, while the images crawled from the Internet have different styles and resolutions in addition to the aforementioned variations.

D. Ablation Studies

To verify the effectiveness of the three losses, we respectively exclude the Fisher loss and the triplet ranking loss of our proposed method and get two incomplete networks, namely, HHN (S-T) and HHN (S-F), where "S", "F" and "T" represent the softmax loss, Fisher loss and triplet ranking loss, respectively. For the first network, only the softmax loss is enforced in the common space. For the second network, the fine-tuning procedure is excluded, and the hash codes are obtained via random projections applied to the floating point features in the common space. We compare the performance of these networks and the whole proposed network in the query-by-image video retrieval task using the selected frames, and the mAPs are shown in Table III. The results show that the separability of the common space, which is guaranteed by the softmax loss, serves as a foundation of good performance. The Fisher loss, which takes the intra-class variations and the inter-class variations into account, enhances the discriminative power of the common space and results in better retrieval performance. Finally, the triplet



(a) (b)

Fig. 6. mAPs of query-by-image video retrieval on the Celebrity-1000 dataset achieved by (a) networks with different β and fixed λ and (b) networks with different λ and fixed $\beta = 0.01$.

ranking loss is critical to generate compact and effective hash representations for retrieval. In the fine-tuning procedure, the optimization of the whole network is beneficial to the optimal compatibility, namely, optimal hash representations.

Furthermore, to verify the effectiveness of sampling methods among the two domains in triplet ranking loss, we relax the requirement that the samples in a positive sample pair must belong to different domains. Specifically, we randomly select samples from the same individual's images and videos to get the positive sample pairs and then use the same methods to generate triplets. Finally, we obtain a network named HHN (S-F-T(r)). The mAPs of the HHN (S-F-T(r)) are also shown in Table III. It can be found that HHN (S-F-T(r)) performs worse than HHN (S-F-T) in all cases and even performs worse than HHN (S-F) occasionally; in the optimization process of HHN (S-F-T(r)), many generated triplets are useless to the cross-domain retrieval task. This demonstrates that our sampling methods are effective and are beneficial to further reduce the discrepancy between the face image and video domains.

E. Parameter Analysis

In our method, α, β and λ are three critical hyper-parameters. α and β in Eq.(9) serve the weight of the softmax loss, the Fisher loss, and the triplet ranking loss, while λ balances the convexity term and the other two terms in the Fisher loss. Since the three losses are utilized in different training stages, and in the pre-training procedure only the softmax loss and the Fisher loss is utilized, we set α as 1 all the time and conduct two experiments on the Celebrity-1000 dataset using the selected frames to verify the sensitiveness of β and λ. In the experiments, we perform the pre-training procedure and obtain the hash codes via random projections applied to the floating point features in the common space.

In the first experiment, we fix λ to 0.01 and vary β from 0 to 1 to train different networks. The mAPs of query-by-image video retrieval are shown in the Fig. 6(a). It can be found that without the Fisher loss, the network cannot achieve satisfactory performance. The performance of our networks remains relatively stable across a range of β. In the second experiment, we fix β and vary λ from 0 to 1 to train different networks. The mAPs are shown in Fig. 6(b). We observe that when the λ is set as a large number *i.e.*, 0.1 or 1, the mAPs will be significantly reduced. When λ is set as a smaller number, the mAPs of our networks remain relatively stable.

TABLE VI
THE ENCODING TIME FOR DIFFERENT ALGORITHMS WHEN THE LENGTH OF THE HASH CODES IS 64

Type	Methods	Image Encoding (ms)	Video Encoding (ms)	
			ICT-TV	Celeb1K
single modality	LSH [26]	0.028	0.117	0.107
	RR [30]	0.024	0.081	0.050
	ITQ [30]	0.035	0.090	0.054
	SH [29]	1.703	5.400	2.543
	SSH [32]	0.018	0.061	0.037
	KSH [33]	0.366	0.593	0.515
	SITQ [30]	0.052	0.141	0.082
	DSH [38]	0.147	0.218	0.268
multiple modality	CCA [30]	0.023	0.023	
	CMSSH [47]	0.045	0.045	
	PDH [46]	0.652	0.652	
	MMNN [49]	0.545	0.545	
	SCM [51]	0.022	0.022	
	HER [25]	10.217	10.217	
Our HHN		**0.279**	**0.344**	

F. Efficiency Study

In this subsection, we analyze the efficiency of our HHN in fully evaluating the effectiveness of the HHN for cross-domain face retrieval. The time cost for cross-domain face retrieval consists of two parts, the encoding time and the retrieval time. The encoding time refers to the time to generate the query hash code for the query face image or video using the trained model, while the retrieval time is the time to get the feedback samples from the database using the generated query hash code. In what follows, we analyze the encoding time and the retrieval time of the proposed HHN. All the experiments are performed on a server with an Intel Core i7-4930K CPU and an NVIDIA Titan X GPU. The DSH [38] and our HHN are implemented using the Caffe tool. Other methods and the retrieval process are implemented using the Matlab. All the reported time costs are the averages of 10 round of tests.

We compare the encoding time cost for image or video of the HHN with the state-of-the-art when the length of the hash codes is 64; the results are listed in Table VI. It should be noted that in our experiments above, to use single-modality hashing methods for cross-domain face retrieval, the face video is regarded as a set of face frames, and the distance between a face image and a face video is computed by averaging the distances between the image and all the frames of the video. We exploit this setting to evaluate the performance of the single-modality hashing methods, and this setting is acceptable since the biggest dataset we utilized in our experiments contains no more than 10000 videos or images. However, this setting is not practical in large-scale cross-domain face retrieval. Thus, we exploit the hard-voting method to obtain the hash codes for videos when computing the video encoding time cost of single-modality hashing methods. For the ICT-TV dataset, the average number of frames for all the videos in the selected subset is about 45, while the average number of frames for the Celeb1K dataset is about 16. Thus, for single-modality hashing methods, we provide the video encoding times for the two datasets in Table VI. Since matrix multiplication is more efficient than loop, it is shown in Table VI that the image encoding time cost and the video encoding time cost for the two datasets are not linearly related. From the table, we find that most traditional hashing methods are faster than deep hashing

methods (HHN, DSH [38] and MMNN [49]) and kernel-based hashing methods (KSH [33] and HER [25]). In addition, the PDH [46], which utilizes the SVM to predict the value of each hash bit, and the SH [29], which compute the hash codes bit by bit rather than using an overall projection matrix, are much slower than other traditional hashing methods.

For all the hashing methods, we use the hash code ranking as the search strategy to compare the query with each sample in the database by rapidly computing their distance and return the samples in ascending order according to distance. Thus, for a database that contains N samples and has hash codes of length L, the space cost of the database is $N * L$ bits and the time complexity of the retrieval process is $O(N * L + N * log(N))$, in which the first term is for computing distance and the second term is for sorting distances when the QuickSort algorithm is utilized, which is the default sorting algorithm in Matlab. In particular, when L is 64 and N is 1,000,000, the space cost of the database is 8 MB. The encoding time of our HHN is 0.279 ms for an image and 0.344 ms for a video. The time cost of computing the distances and sorting the distances is 29.839 ms and 79.558 ms, respectively. It should be noted that the distance computing process also benefits from matrix multiplication.

From the above analysis, we find that although our HHN is more time consuming than most traditional hashing methods in generating the query hash code, in the face retrieval process, the retrieval time cost is almost two orders of magnitude larger than the encoding time cost. And the gap between them will be even larger in practical applications in which the database contains billions of samples. As a result, the HHN is still scalable since the time cost of retrieval is almost as fast as traditional hashing methods while outperforming other methods.

V. Conclusion

In this paper, we have proposed a heterogeneous network for face retrieval across image and video domains. The heterogeneous hashing network containing an image branch and a video branch is able to generate isomorphic compact hash representations of both face images and face videos from heterogeneous spaces. The network is trained with Fisher loss, softmax loss, and triplet ranking loss. The Fisher loss, which uses the difference form of the within-class and the between-class scatter, is feasible for the mini-batch based optimization method. The heterogeneous hashing network provides a general framework for deep learning-based cross-domain hashing methods and can be easily adopted in many other cross-domain retrieval tasks. Experiments on a large-scale face video dataset and two challenging TV-series datasets show that the proposed method is effective in face retrieval across image and video domains.

References

[1] O. Arandjelovic, G. Shakhnarovich, J. Fisher, R. Cipolla, and T. Darrell, "Face recognition with image sets using manifold density divergence," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 1, pp. 581–588.

[2] G. Shakhnarovich, J. W. Fisher, and T. Darrell, "Face recognition from long-term observations," in *Proc. Eur. Conf. Comput. Vis.*, 2002, pp. 851–865.

[3] H. Cevikalp and B. Triggs, "Face recognition based on image sets," in *Proc. IEEEComput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2567–2573.

[4] Y. Hu, A. S. Mian, and R. Owens, "Sparse approximated nearest points for image set classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 121–128.

[5] T.-K. Kim, J. Kittler, and R. Cipolla, "Discriminative learning and recognition of image set classes using canonical correlations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1005–1018, Jun. 2007.

[6] R. Wang, H. Guo, L. S. Davis, and Q. Dai, "Covariance discriminative learning: A natural and efficient approach to image set classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2496–2503.

[7] R. Vemulapalli, J. K. Pillai, and R. Chellappa, "Kernel learning for extrinsic classification of manifold features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1782–1789.

[8] L. Wang, J. Zhang, L. Zhou, C. Tang, and W. Li, "Beyond covariance: Feature representation with nonlinear kernel matrices," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4570–4578.

[9] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Web-scale training for face identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2476–2574.

[10] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 815–823.

[11] Y. Sun, X. Wang, and X. Tang, "Sparsifying neural network connections for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4856–4864.

[12] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis*., 2015, pp. 4489–4497.

[13] J. Yang *et al.*," Neural aggregation network for video face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5216–5225.

[14] H. Lee, Y. Chung, J. Kim, and D. Park, "Face image retrieval using sparse representation classifier with Gabor-LBP histogram," in *Proc. Int. Conf. Inf. Secur. Appl.*, 2010, pp. 273–280.

[15] Z. Wu, Q. Ke, J. Sun, and H.-Y. Shum, "Scalable face image retrieval with identity-based quantization and multireference reranking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 10, pp. 1991–2001, Oct. 2011.

[16] D. Wang *et al.*, "Retrieval-based face annotation by weak label regularized local coordinate coding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 550–563, Mar. 2014.

[17] M. Kafai, K. Eshghi, and B. Bhanu, "Discrete cosine transform locality-sensitive hashes for face retrieval," *IEEE Trans. Multimedia*, vol. 16, no. 4, pp. 1090–1103, Jun. 2014.

[18] B.-C. Chen, C.-S. Chen, and W. H. Hsu, "Face recognition and retrieval using cross-age reference coding with cross-age celebrity dataset," *IEEE Trans. Multimedia*, vol. 17, no. 6, pp. 804–815, Jun. 2015.

[19] Y. Li *et al.*, "Two birds, one stone: Jointly learning binary code for large-scale face image retrieval and attributes prediction," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3819–3827.

[20] D. Wang, C. Otto, and A. K. Jain, "Face search at scale," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1122–1136, Jun. 2017.

[21] B. C. Chen, Y. H. Kuo, Y. Y. Chen, K. Y. Chu, and W. Hsu, "Semi-supervised face image retrieval using sparse coding with identity constraint," in *Proc. 19th ACM Int. Conf. Multimedia*, 2011, pp. 1369–1372.

[22] Y. Li, R. Wang, Z. Cui, S. Shan, and X. Chen, "Compact video code and its application to robust face retrieval in TV-series," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–12.

[23] Y. Li, R. Wang, Z. Cui, S. Shan, and C. Xilin, "Spatial pyramid covariance-based compact video code for robust face retrieval in TV-series," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5905–5919, Dec. 2016.

[24] Z. Dong, S. Jia, T. Wu, and M. Pei, "Face video retrieval via deep learning of binary hash representations," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 3471–3477.

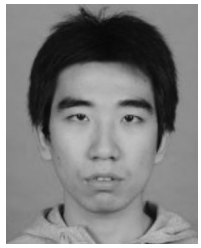
[25] Y. Li, R. Wang, Z. Huang, S. Shan, and X. Chen, "Face video retrieval with image query via hashing across Euclidean space and riemannian manifold," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4758–4767.

[26] A. Gionis *et al.*, "Similarity search in high dimensions via hashing," in *Proc. 25th Int. Conf. Very Large Data Bases*, 1999, vol. 99, pp. 518–529.

[27] B. Kulis and K. Grauman, "Kernelized locality-sensitive hashing for scalable image search," in *Proc. IEEE 12thInt. Conf. Comput. Vis.*, 2009, pp. 2130–2137.

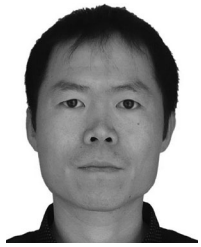
[28] M. Raginsky and S. Lazebnik, "Locality-sensitive binary codes from shift-invariant kernels," in *Proc. 23rd Annu. Conf. Neural Inf. Process. Syst.*, 2009, pp. 1509–1517.

[29] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proc. Neural Inf. Process. Syst.*, 2009, pp. 1753–1760.
[30] Y. Gong and S. Lazebnik, "Iterative quantization: A procrustean approach to learning binary codes," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 817–824.
[31] X. Liu *et al.*, "Multilinear hyperplane hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5119–5127.
[32] J. Wang, S. Kumar, and S.-F. Chang, "Semi-supervised hashing for scalable image retrieval," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3424–3431.
[33] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang, "Supervised hashing with kernels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., 2012, pp. 2074–2081.
[34] F. Cakir and S. Sclaroff, "Adaptive hashing for fast similarity search," in *Proc. Proc. IEEE Int. Conf. Comput. Vis*., 2015, pp. 1044–1052.
[35] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, "Supervised hashing for image retrieval via image representation learning," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 2156–2162.
[36] F. Zhao, Y. Huang, L. Wang, and T. Tan, "Deep semantic ranking based hashing for multi-label image retrieval," in *Proc. IEEE Comput. Vis. Pattern Recognit*., 2015, pp. 1556–1564.
[37] K. Lin, J. Lu, C.-S. Chen, and J. Zhou, "Learning compact binary descriptors with unsupervised deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1183–1192.
[38] H. Liu, R. Wang, S. Shan, and X. Chen, "Deep supervised hashing for fast image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2064–2072.
[39] W.-J. Li, S. Wang, and W.-C. Kang, "Feature learning based deep supervised hashing with pairwise labels," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 1711–1717.
[40] V. E. Liong, J. Lu, Y.-P. Tan, and J. Zhou, "Deep video hashing," *IEEE Trans. Multimedia*, vol. 19, no. 6, pp. 1209–1219, Jul. 2017.
[41] Q.-Y. Jiang and W.-J. Li, "Deep cross-modal hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3270–3278.
[42] Z. Lin, G. Ding, M. Hu, and J. Wang, "Semantics-preserving hashing for cross-view retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3864–3872.
[43] Y. Cao, M. Long, J. Wang, Q. Yang, and P. S. Yu, "Deep visual-semantic hashing for cross-modal retrieval," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1445–1454.
[44] F. Feng, X. Wang, and R. Li, "Cross-modal retrieval with correspondence autoencoder," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 7–16.
[45] S. Kumar and R. Udupa, "Learning hash functions for cross-view similarity search," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, vol. 22, no. 1, pp. 1360–1365.
[46] M. Rastegari, J. Choi, S. Fakhraei, H. Daumé III, and L. S. Davis, "Predictable dual-view hashing," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 1328–1336.
[47] M. M. Bronstein, A. M. Bronstein, F. Michel, and N. Paragios, "Data fusion through cross-modality metric learning using similarity-sensitive hashing," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2000, vol. 1, no. 2., pp. 3594–3601.
[48] Y. Zhen and D.-Y. Yeung, "A probabilistic model for multimodal hash function learning," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 940–948.
[49] J. Masci, M. M. Bronstein, A. M. Bronstein, and J. Schmidhuber, "Multimodal similarity-preserving hashing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 824–830, Apr. 2014.
[50] D. Zhai *et al.*, "Parametric local multimodal hashing for cross-view similarity search," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 2754–2760.
[51] D. Zhang and W.-J. Li, "Large-scale supervised multimodal hashing with semantic correlation maximization," in *28th AAAI Conf. Artif. Intell.*, 2014, vol. 1, no. 2, pp. 2177–2183.
[52] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Fisher discrimination dictionary learning for sparse representation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 543–550.
[53] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian detection via classification on riemannian manifolds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1713–1727, Oct. 2008.
[54] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia.*, 2014, pp. 675–678.
[55] Y. Li, R. Wang, S. Shan, and X. Chen, "Hierarchical hybrid statistic based video binary code and its application to face retrieval in TV-series," in *Proc. 11th IEEE Int. Conf. Workshops Autom. Face Gesture Recognit.*, 2015, pp. 1–8.
[56] L. Liu, L. Zhang, H. Liu, and S. Yan, "Toward large-population face identification in unconstrained videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 11, pp. 1874–1884, Nov. 2014.
[57] R. Xu *et al.*, "Binary subspace coding for query-by-image video retrieval," 2016, *arXiv:1612.01657*.
[58] X. Zhu *et al.*, "Learning heterogeneous dictionary pair with feature projection matrix for pedestrian video retrieval via single query image," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4341–4348.
[59] Z. Chen, W. Zhang, B. Deng, H. Xie, and X. Gu, "Name-face association with web facial image supervision," *Multimedia Syst.*, pp. 1–20, 2017.
[60] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016.



Chenchen Jing received the B.S. degree in computer science in 2016 from the Beijing Institute of Technology, Beijing, China, where he is currently working toward the Ph.D. degree. His research interests include computer vision, pattern recognition, and face recognition.

Zhen Dong received the B.S. degree in June 2011 from Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing, China, where he has been working toward the Ph.D. degree under the supervision of Prof. Yunde Jia since September 2011. His research interests include computer vision, machine learning, and face recognition.



Mingtao Pei received the Ph.D. degree in computer science from Beijing Institute of Technology, Beijing, China, in 2004. From 2009 to 2011, he was a Visiting Scholar with the Center for Image and Vision Science, University of California, Los Angeles, CA, USA. He is currently an Associate Professor with the School of Computer Science, Beijing Institute of Technology. His main research interest includes computer vision with an emphasis on event recognition and machine learning. Prof. Pei is a member of the China Computer Federation.

Yunde Jia (M'11) received the B.S., M.S., and Ph.D. degrees from the Beijing Institute of Technology (BIT), Beijing, China, in 1983, 1986, and 2000, respectively. From 1995 to 1997, he was a Visiting Scientist with the Robot Institute, Carnegie Mellon University, Pittsburgh, PA, USA. He is currently a Professor with the School of Computer Science, BIT, and the Team Head with the BIT innovation on vision and media computing. He is the Director with the Beijing Laboratory of Intelligent Information Technology, Beijing, China. He has authored and coauthored more than 300 publications in computer vision and media computing. In recent years, his interests have extended to vision-based HCI and HRI, intelligent robotics, and cognitive systems.