Hybrid Video and Image Hashing for Robust Face Retrieval

Ruikui Wang[1,2], Shishi Qiao[1,2], Ruiping Wang[1,2], Shiguang Shan[1,2], Xilin Chen[1,2]
[1] Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, 100190, China
[2] University of Chinese Academy of Sciences, Beijing, 100049, China
{ruikui.wang, shishi.qiao}@vipl.ict.ac.cn, {wangruiping, sgshan, xlchen}@ict.ac.cn

***Abstract*— Video face retrieval (VFR) is an appealing and practical computer vision task, which aims to search particular character from masses of videos like in TV-Series. The challenges of this task mainly lie in two aspects, i.e. faces in such videos contain complex appearance variations with uncontrollable shooting environment and searching from big data usually requires high efficiency in both space and time. To fulfill this task, current works typically proceed in a learning to hash manner by fusing single-frame features within a video to obtain the video representation and further embedding it into Hamming space to yield video binary codes. The feature fusion stage has inevitably discarded too much frame information and leads to less discriminative video codes. In this paper, we propose Hybrid Video and Image Hashing (HVIH) to learn more effective binary codes for face videos. Specifically, we fully exploit the dense frame features rather than simply discarding them after the video level fusion and jointly optimize binary codes for the video and its composed frames in adapted supervised manners. To achieve more robust video representation, we introduce a module of video center alignment to ensure the binary codes location of the video and its frames to be as compact and consistent as possible in the Hamming space, which naturally facilitates both tasks of video-to-video retrieval and image-to-video retrieval. Extensive experiments on two challenging video face databases demonstrate the superiority of our approach over the state-of-the-art.**

I. INTRODUCTION

Given one video track of some specific characters, video face retrieval aims to search shots containing the corresponding person [34], as demonstrated in Fig. 1. It is a promising research topic with gradually increasing attention. Technically speaking, video face retrieval deals with video understanding and data retrieval technology, both of which are crucial for computer vision. This task has a wide range of potential applications, such as: 'only look him/her' - where the video can be automatically skipped to the corresponding shot containing the specific character; retrieve all clips containing a particular family member from thousands of short videos [35]; and fast spotting and tracking criminal suspects from masses of surveillance videos.

Intriguingly, the basic technique for video face retrieval is face recognition, which has made remarkable progress with the promotion of deep learning in recent years [27], [10], [16]. Compared with still-face retrieval, video face retrieval has its unique characteristics. Specifically, one video tends to contain rich variations caused by illumination, pose, expressions, resolution and occlusion [34], [3]. Therefore, how to integrate these complementary information from single

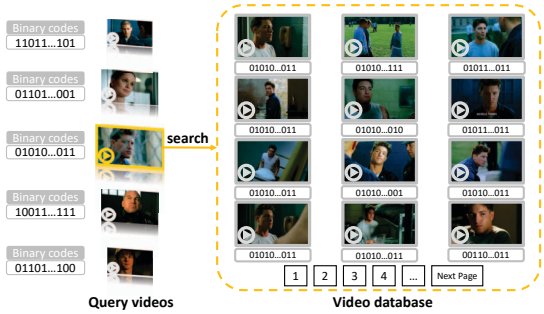


Fig. 1. A schematic diagram of video face retrieval. With a query of a desired celebrity's video clip, all shots relevant with him/her in database are retrieved and ranked according to their similarities to the query.

frames to obtain a comprehensive and robust representation for the video needs to be considered adequately.

There have been substantial efforts devoting to fuse dense frame features to represent a whole video and then embed it into Hamming space leveraging hashing function [24], [25], [31], [11]. However, most of them obtain video representation by fusing single-frames features before hashing layer, which means that single-frame information has been early discarded after feature fusion and cannot be optimized by supervised signals directly. Nevertheless, the CNN feature of each frame provides very helpful information for distinguishing videos from different identities, which has been totally ignored by above scheme. Besides, treating video as an isolated point in Hamming space lacks of robustness especially when the variations are large within one video. In this paper, we propose Hybrid Video and Image Hashing (HVIH) to alleviate the above problem and yield robust video codes. Specifically, we embed both video-level and frame-level real-valued features into Hamming space and then distinguish each frame from multiple identities, which aims at exploiting dense frames to guide video modeling and roughly determine different video's location in Hamming space. After such procedure, the video-level binary codes can be obtained by temporal feature pooling. Because of the retention of frame-level binary codes, our scheme naturally facilitates both tasks of video-to-video retrieval and image-to-video retrieval. Based on the roughly location determined by dense frame information, we further increase the video codes' discriminability, which is crucial for video-level re-

978-1-7281-3079-8/20/$31.00 ©2020 IEEE
DOI 10.1109/FG47880.2020.00028

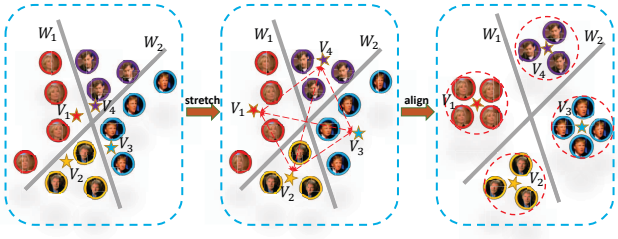


Fig. 2. An illustration of video representation learning procedure. The pentagram denotes the video and face motifs denote its composed frames. Each color denotes one character.

trieval, by combining metric learning techniques [40], [46]. Considering the robustness of video codes and the ability of image-to-video retrieval, we rectify the location of video and composed frames in Hamming space as compact and consistent as possible. Fig. 2 gives an intuitive understanding for this procedure.

To verify the validity of the proposed method, we conduct video face retrieval experiments on two datasets. The results show the effectiveness of the proposed HVIH against state-of-the-arts retrieval methods.

II. RELATED WORK

Recent years have witnessed more and more investigation on video face retrieval [34], [35], [24], [1], [2], [25], [11], [13], [12], [18]. For example, [1], [2] proposed a cascade of processing steps to normalize the effects of the changing image environment and used the signature image to retrieval a face shot. [8] investigated the identification problem for face clips of TV-Series; [5] leveraged fan transcripts and subtitles to achieve person identification in TV-Series. Actually, most of these previous works tried to construct an end-to-end system, including shot boundary detection, face detection, face tracking, and face retrieval, to accomplish comprehensive face video analysis. It is admittedly believed that paramount technical components for this task lie in two aspects, i.e. the video modeling and fast retrieval, which are also the main research subjects in this paper.

Actually, face video can be treated as an image set. Given a face video, for identity retrieval task, we mainly focus on its frame appearance other than temporal motion information [31]. As for image set modeling, holistic modeling approaches empirically have an advantage over processing each frame separately. Traditional representative methods include linear subspaces [20], affine subspaces [7], bilinear matrices [24], [25], etc. However, these methods mainly rest on hand-crafted features. More recently, CNN based methods gradually exhibit the superiority on many vision tasks, such as video classification [44], [41], video face recognition [43], [29], [33], [47], video face retrieval [24], [25], [31], [11], [12], [18], event detection [41], pose estimation [30]. The core technique of these methods is frame-level feature extraction and aggregation. The main shortage of above holistic modeling schemes is that they discard the frame-level imformation which is beneficial to enhance the robustness of video representation and boost the video retrieval performance when video representation is obtained. The second drawback that lies in most existing methods is high dimensionality which severely limits their applicability in large-scale video retrieval.

Hashing is an efficient tool for large-scale data retrieval. Compared with traditional searching methods [9], hashing has lower time and space complexity. The classical work contains the family of methods known as Locality Sensitive Hashing (LSH) [14] and its variants [32]. It is worth noted that LSHs always need much longer codes to obtain better performance in real searching task. To reduce the major shortcoming, data-dependent hashing methods try to embed either data structure or semantic similarity to compact binary codes. These methods can be further partitioned into unsupervised and (semi-) supervised methods. Important unsupervised methods include Spectral Hashing (SH) [39], Iterative Quantization (ITQ) [15], K-Nearest Neighbors Hashing (KNNH) [17], etc. Without any label information, unsupervised methods are usually inferior to supervised methods. Representative (semi-) supervised methods include Semi-Supervised Hashing (SSH) [37], Kernel-based Supervised Hashing (KSH) [28], etc. Please see [38] for a more comprehensive survey. More recently, deep hashing methods become more and more popular. Simultaneously learning image feature and hash function in a end-to-end manner is the key to deep hashing over traditional methods. Most deep hashing methods are similar in feature extracting module, hash function learning module reflects difference instead. Representative methods include DSH [26], DPSH [23], DNNH [22], DSRBH [45], HashNet [6], SSDH [42], Greedy Hash [36]. Thanks to the extensive application of hashing, binary based VFR methods gradually attract the attention of researchers. Representative methods include [24], [31], [18]. These works make video face retrieval more fast and practical.

III. PROPOSED METHOD

In order to efficiently search specific character by a query video from masses of databases, what we need firstly is to get a robust and powerful representation for each face video. Actually, a face video can be considered as a set of face frames containing various head poses and appearance. How to mine complementary information from these frames and jointly optimize the feature extraction and fusion procedure is the core of the face video representation.

A. Hybrid Video and Image Hashing

For this target, we propose HVIH as illustrated in Fig. 3. Let $V = \{I_1, I_2, ..., I_N\}$ be a face video with N frames, where I_i denotes the i-th frame. We firstly propagate each frame through the stacked convolution network $\mathcal{F}$ and then the deep features for each frame are obtained as $f_i = \mathcal{F}(I_i), f_i \in R^d$.

Secondly, we leverage a compact embedding model, which maps a high-dimensional floating feature to a succinct representation which lies in Hamming space.

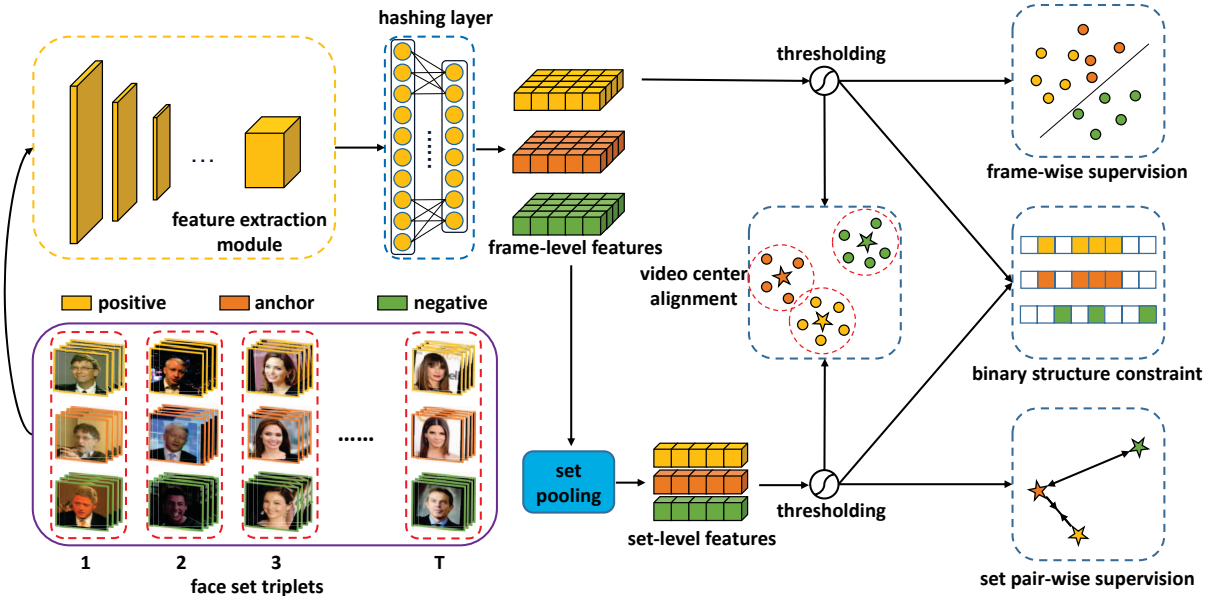


Fig. 3. Illustration of the whole framework of the proposed method. The input of the model is batches composed of set triplets. Each set triplet passes through the feature extraction module to generate hidden representations, which will be fed to hashing layer to generate frame-level binary codes and set-level binary codes. Then two different supervised manner are devised for two types of binary codes accordingly. The anchors, positive instances and negative instances are denoted by different colors. In addition, the dot denotes frame-level binary codes and pentagram denotes set-level binary codes.

At last, considering the scalability of the video representation and comparability of videos with different number of frames, we get fixed dimension representation by using temporal feature pooling. Temporal feature pooling has been widely used for video face retrieval and video classication [31], [44], the output vector of which can be used to achieve video-level retrieval. Above all, the HVIH can be formulated as follows:

$$b_i^V = \sigma(W^T f_i), W \in R^{d\times K}, b_i^V \in R^K \tag{1}$$

$$b^V = \sigma(\frac{1}{N}\sum_{i=1}^{N} W^T f_i), b^V \in R^K \tag{2}$$

The W in (1) denotes hashing function, which acts as the compact embedding module mentiond above. $\sigma(\cdot)$ denotes the threshold function, which quantizes real-value feature to be binary. The b^V and b_i^V denote the binary codes of the video and its composed i-th frame respectively. Considering the optimization in an end-to-end manner, we relax the binary constraints to range constraints of [0,1], i.e. use $sigmoid$ to approximate sgn as threshold function. In order to reduce the quantitative loss caused by this approximation, we conduct additional binary regularizations on the thresholded outputs introduced in Section III-B.

B. BINARY CODE LEARNING

In terms of a coming face video, we can get compact binary-like representation with relaxation mentioned in previous section. As we all known, there are two advantages leveraging binary features, i.e. the concise space demand, the low time cost. These two points are very helpful for fast large-scale video face retrieval. However, the HVIH formulated as above does not incorporate any supervision information, which will definitely enhance the retrieval accuracy. To solve this problem, we drive the discriminative binary codes learning in two different supervised manners respect to video and frame. Next we will dicuss how to optimize these two different representations guided by the provided identity signals.

1) Frame-wise Supervision: In previous representation learning process, we have preserved the frames in the Hamming space. The goal is to make these frames discriminative by imposing identity constraints on them. There is no doubt that the discriminability of binary codes in Hamming space is crucial most, i.e. each frame can be distinguished from multiple identities, which has an impact on semantic retrieval. For another, once these frames become much discriminative, they will support the videos' surroundings in Hamming space and guarantee the correct direction of video discriminative learning in future steps. It is so-called supported frames. Formally, let b_j^i denotes the binary code of the j-th frame of the i-th video instance. Then frame-wise supervision L_{frame} can be formulated as,

$$L_{frame} = \frac{1}{M}\sum_{i=1}^{M}\left\{\frac{1}{|V_i|}\sum_{I_j\in V_i}\frac{exp(\theta_{c^i}^T b_j^i)}{\sum_{c=1}^{C} exp(\theta_c^T b_j^i)}\right\} \tag{3}$$

where M is the number of video instances in a batch, C is the number of person identities, V_i denotes i-th video instance and $|V_i|$ indicates the number of frames in V_i, θ_{c^i} denotes c^i-th hashing hyperplane and c^i indicates that the i-th video instance belongs to c^i-th person identity.

2) Set Pair-wise Supervision: Once we differentiated supported frames from each identity, it is equivalent to establishing the image set partition in Hamming space. According to

our video modeling scheme, a video point tends to locate in the center region of a image set partition. It merits attention that if we utilize the same supervised manner for video, it may have limited contribution on further boosting the model's integral discriminability, which results from that most of video cases have satisified the hashing hyperplane partition principle. To address this problem, we turn to metric learning techniques. Considering image set as an entirety, we combine two types of discriminability constraint, i.e. intra-class compactness and inter-class separability. Specifically, the codes of semantically similar videos should be as close as possible, while the codes of dissimilar videos being far away. Formally, let a, p, n as a set triplet (Actually, a triplet can be understood as a set of pairs) where a denotes the anchor, and p, n denote positive and negative samples, then their distance correlations can be formulated as,

$$L_{video} = \frac{1}{|T|} \sum_{(a,p,n)\in T} \beta \cdot max(d_H(b^a, b^p) - d_H(b^a, b^n) + m, 0) + \gamma \cdot d_H(b^a, b^p) \tag{4}$$

where b^a, b^p, b^n denote K-bit binary codes of video instance in a set triplet and they can be computed using (2). $d_H(\cdot,\cdot)$ denotes Hamming distance between two binary vectors, $|T|$ denotes the scale of set triplets, $m > 0$ denotes margin threshold parameter and β, γ denote balancing coefficient of the two loss terms.

3) Video Center Alignment (VCA): Until now, we have access to optimized binary codes of the face video and their composed frames. However, the set pooling is occured in the same-dimension real-valued space before thresholding, which may cause that the video shifts from the center of their composed frames especially when large variations occur within the video. In order to get robust video representation that can leverage the frame-wise variations, we introduce a video center alignment module to rectify the video's location to make images and videos constrain each other. Please see Fig. 2 for more intuitive understanding. Formally, the video center alignment module can be imposed as follows:

$$L_{vca} = \frac{1}{M} \sum_{i=1}^{M} \left\| b^{V_i} - \frac{1}{|V_i|} \sum_{I_j \in V_i} b_j^{V_i} \right\|_p \tag{5}$$

where arbitraty norm can be assigned by setting the subscript p. In our experiments, we choose the L2-norm (i.e. $p = 2$).

It is worth noting that VCA not only enhances the power of video representation but also endows the ability of retrieval across image and video. Besides, VCA is only used during training phase. When new test videos coming, we only need to propagate their frames or key frames after sampling through the stacked convolution network and directly execute set pooling in a scalable way to get videos' binary codes.

4) Binary Structure Constraint: It would be advisable if (3) and (4) can be optimized directly with binary constraint, whereas it is nonviable because getting binary codes needs to threshold the model outputs with sgn operator. This will bring trouble to the network training with back propagation algorithm. To alleviate this issue, the general process is to relax the strict binary constraint using $sigmoid$ operator as the threshold function. Nevertheless, working with such non-linear functions would precipitate suboptimal binary codes due to the heterogeneity between the continuous real-valued space and Hamming space. As a compensation, we impose a binary structure constraint term on thresholded features to reduce the approximation cost. To be specific, we force the network outputs to approach the desired discrete values $(0/1)$ as well as maximize the variance of each hash bit, which is motivated by [37], [42] and formulated as:

$$L_{bsc} = \frac{1}{M} \sum_{i=1}^{M} \left\{ \frac{1}{2\,|V_i|} \sum_{I_j \in V_i} \left(\left\| mean(b_j^i) - 0.5 \right\|_2^2 - \left\| b_j^i - 0.5I \right\|_2^2 \right) \right\} \tag{6}$$

In pratice, we also impose such binary structure constraint on binary codes of videos. After above narrations, our overall loss function can be reached by combining (3), (4), (5) and (6):

$$L = L_{frame} + \lambda_1 \cdot L_{video} + \lambda_2 \cdot L_{vca} + \lambda_3 \cdot L_{bsc} \tag{7}$$

C. Implementation Details

Network parameters. We implement our method with Caffe platform[1]. For fair comparison, we use the same backbone for comparative methods. In this paper, our backbone consists of four doubled-convolution followed max-pooling layers and one doubled-convolution followed average-pooling layer. The convolution layers include (64, 32, 16, 8 and 4)×2 3×3 filters with stride 1 respectively, the size of max-pooling window is 2×2 with stride 2, the size of average-pooling window is 4×4 with stride 1. In practice, the backbone can be any type of deep architecture. This is not the focus of our discussion here. Following the frame-level feature extraction module, we achieve hash layer by several fully connected layers. Then, we impose the loss functions on the feature after threshold function $sigmoid$. Besides, all the convolution layers are followed by the ReLU activation function.

Before the training phase, we initilize hash layer with "Gaussian" method and corresponding initial variance is set as 0.01. The other parametric layers are initialized with "Xavier" method. We set batch-size as 240 (the video number in a batch is 24, i.e. M=24), momentum as 0.9, and weight-decay as 0.0005. The model is trained with 100,000 iterations. The learning rate is set as 0.001 and decreased by a tenth every 50,000 steps.

Training methodology. To speed up the training process and better use of storage space, we construct each mini-batch formed by set triplets in a off-line way. More concretely, we randomly take out $2 * M/3$ characters as the candidate pool for each mini-batch in the first step. Then anchor person IDs and positive person IDs are selected from former half

[1]Our source codes are available at http://vipl.ict.ac.cn/resources/codes.

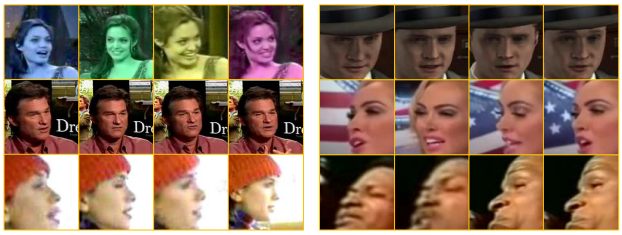


Fig. 4. Face clip examples of the two large scale video face datasets, where left four columns are sampled from YTC while right four columns are sampled from UMDFaces. Frames in the same row are from one video.

TABLE I

DATA STATISTICS AND DIVISION OF TWO LARGE SCALE VIDEO FACE DATASETS.

Dataset	YouTube Celebrities	UMDFaces
Identities	47	200
Training videos	7,190	6,614
Testing videos	3,101	3,422
Training frames	206,195	189,051
Testing frames	89,234	97,735

of the candidate pool while negative person IDs are selected from the other half of the pool, in such way, $M/3$ triplets can be obtained in each mini-batch. Thirdly, for each triplet, we randomly select one video according to its ID as triplet element, so there are M videos in one mini-batch. At last, since the frame count of video is different, for each video, 10 frames are randomly selected from the whole video to form a set. There are two major advantages to constructing a mini-batch in this way: firstly, the offline configuration save the time overhead of online enumeration; Imagining that above complicated sample scheme is implemented online, the training efficiency will be greatly reduced. Secondly, the distribution of each ID will be relatively uniform with the candidate pool. Another trick in training phase is the use of finetuning. Specifically, the feature extraction module is fixed while the hash layer and subsequent parameter layers are trained in first stage. Then the whole network will be finetuned in a decay learning rate in the second stage.

IV. EXPERIMENTS

A. Datasets

We conduct experiments on two large scale video face databases : YouTube Celebrities (YTC) [19] and UMDFaces [4]. YTC is a widely investigated and challenging benchmark containing 1,910 videos of 47 characters collected from YouTube. In practice, the final face tracks are parsed from raw videos leveraging several technologies, e.g. shot boundary detection, face detection, tracking, and facial landmark localization. Large variations such as illumination, background, yaw, among shots remarkably increase the challenge for accurate retrieval. The second dataset UMDFaces is a recently released large scale video face benchmark, which contains both still-images and videos. Concretely, the original video settings contains 22,075 raw videos for 3,107 characters (about 7 videos per character). To guarantee the purity of databases, we select a subset with 200 subjects for the experimental evaluation. Some exemplar face tracks of the two video databases are shown in Fig. 4. The statistical information and splits of two datasets can be found in Table I.

B. Experimental Settings

In the experiments, the input resolution of the face is set as 64×64 pixels. The margin m in (5) is empirically set as 2.0. Besides, β, γ and λ_2 are set as 0.1, 0.5 and 0.01, the balance parameters λ_1 and λ_3 are both set as 1.0 without elaborate configuration. For all comparative methods, crucial parameters are carefully tuned according to the recommendations in the original literatures and source codes. For quantitative evaluation, we use two standard criteria, i.e. mean Average Precision (mAP) and the precision recall (PR) curve.

C. Evaluation on Different Video Representation

As discussed in Section III-A, the video modeling is a crucial technical component of video face retrieval. In this subsection, we will validate the effectiveness of the proposed video representation algorithm. We mainly focus on three baselines, i.e. *single-image*, *single-video* and *joint image and video*. Concretely, *single-image* denotes that we treat each video as image set and fuse corresponding frame-level binary codes as final video binary codes. In this baseline, we only leverage frame-wise supervision mentioned in Section III-B. Another noteworthy point is the batch construction strategy, since that there is no need to generate sample pairs or triplets for single-image, we randomly select frames for each batch where the probability of selecting each frame is uniform. *single-video* denotes that we only preserve video-level binary codes in Hamming space, i.e. we only leverage set-pair-wise supervision mentioned in Section III-B. *joint image and video* denotes that we jointly optimize binary codes of video and its composed frames. *joint image and video* ++ denotes that we jointly optimize binary codes of video and its composed frames with VCA module. For *single-video* and *joint image and video*, we both adopt the same batch construction scheme introduced in Section III-C.

The results of different video modeling strategy are listed in Table II. From the comparison, we have made two consistent foundings: 1) Our *joint image and video* performs superiorer than other baselines as expected in most cases. This may partly attribute to our video representation learning scheme, i.e. improving the discriminabilities of frame-level representations is beneficial to boost the robustness of the video representation and set-pair supervision further boosts the discriminability of the video-level binary codes. Notice that our approach is more advanced on the UMDfaces dataset with large appearance variations, which sidely shows that our scheme is more suitable for processing complicated variations in face videos. 2) The performance of *single-image* is better than *single-video*. On one hand, fully discarding the information after video fusion is harmful to the learning of video representations. On the other hand, the scale of

TABLE II

EVALUATION ON DIFFERENT VIDEO REPRESENTATION LEARNING PARADIGMS WITH MAP ON TWO DATASETS. 'JOINT IMAGE AND VIDEO ++' DENOTES RESULTS WITH VCA.

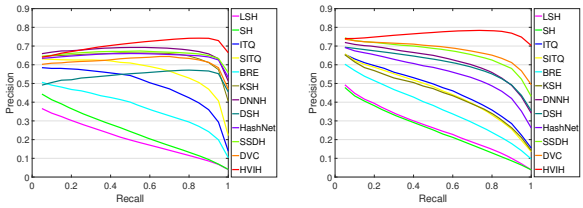
Video Modeling Strategy	YouTube Celebrities				UMDFaces			
	12-bit	24-bit	36-bit	48-bit	12-bit	24-bit	36-bit	48-bit
single-image	**0.5925**	0.6226	0.6451	0.6599	0.4645	0.6064	0.6456	0.6667
single-video	0.5085	0.5230	0.5786	0.5815	0.3165	0.3685	0.3729	0.3800
joint image and video	0.5579	0.6475	0.6804	0.6885	**0.5645**	0.6745	0.7233	0.7568
joint image and video ++	0.5602	**0.6483**	**0.6989**	**0.7042**	0.5570	**0.6846**	**0.7398**	**0.7628**

available training samples for frames is much larger than that of videos, which further leads to the huge performance difference. We can also note that, in some specific bits on YTC, *single-image* even surpasses *joint image and video*. Actually, *single-image* aims to optimize binary codes for each frame and aggregate them to form an integrated video code, which may work well when face variations among frames are relatively small.

D. Comparison with the State-of-the-art Hashing Methods

To demonstrate the effectiveness of the proposed method, we compare it with several binary code learning methods, including LSH [14], SH [39], ITQ [15], SITQ [15], BRE [21], KSH [28], DNNH [22], DSH [26], DVC [31], HashNet [6] and SSDH [42]. For fair comparison, we adopt totally the same backbone with comparative deep hashing methods and extract corrspondingly deep feature for all traditional hashing methods. Please note that most of the competitive hashing methods except DVC cannot apply for VFR task directly. Following DVC, we reproduce them by regarding each face frame as a sample and fuse these frame-level binary codes to obtain video-level binary codes by hard-voting.

Table III and Fig. 5 give the comparison performance in mAP and PR curve. Overall, we have the following three observations: 1) Deep hashing methods generally achieve higher retrieval performance than traditional hashing methods with deep features on two datasets, which suggests that simultaneously learning image feature and optimizing hash function in an end-to-end manner is predominant. 2) Compared with state-of-the-art deep hashing methods, ours achieves the highest performance in most cases and especially superior on the UMDFaces dataset which contains more characters and complex appearance variations than YTC. A possible interpretation is our model leverages dense-frame supervision and video supervision jointly, which results in larger model capacity. We can also review this phenomenon from hashing perspective. The performance differences among other deep hashing methods mainly come from the their supervised manners. DSH and HashNet adopt pair-wise supervision while DNNH adopts triplet-wise supervision. SSDH belongs to point-wise supervision and achieves generally more excellent performance than former two types. Our method imposes more reasonable supervised manner on different modality, i.e. video and frame, which belongs to a hybrid supervised manner. 3) As for DVC, it optimizes smooth upper bound on triplet loss which leads to stable



(a) (b)

Fig. 5. Comparison of precision recall curve on two databases, i.e. (a) YouTube Celebrities (YTC) and (b) UMDFaces. Without loss of generality, we show the results with 48-bit for a demonstration.

convergence and obtain better performance than pair-wise and triplet-wise methods. However, it is still inferior to our method. We attribute it to our sufficient exploitation of dense frames in Hamming space.

E. Evaluation on Cross-modality Retrieval

As pointed before, our method can also deal with image-to-video retrieval. In order to verify the superiority of our method under this protocol, we compare with several competitive retrieval methods. Specifically, for all comparative methods, we sample one frame from each testing video to construct new query set and original training videos as database. The comparison results are listed in Table IV. We can see that our method significantly outperforms other methods in most cases. This is mainly due to the fact that our method optimizes frame-level binary codes in Hamming space and align them with videos such that the learned codes can be applied to this cross-modality retrieval task. It is worth noting that DVC cannot obtain frame-level binary codes, and thus it cannot apply for image-to-video retrieval.

F. Parameter Sensitivity Analysis of Video Center Alignment

As discussed in III-B, we aim to rectify the location of video and composed frames in Hamming space as compact and consistent as possible with VCA. From the last two lines of Table II, we can see that VCA authentically boosts the retrieval performance in most cases. In this subsection, we analysis the parameter sensitivity of VCA, i.e. λ_2. Without loss of generality, we only test the case under 48-bit. More specifically, we fix all other unrelated parameters, only research the performance variation tendencies of λ_2. Fig. 6 shows that our method achieves better performance than baseline (i.e. setting without VCA) when λ_2 varies over a relatively large range. In other words, our method is not sensitive to λ_2 in a large range.

TABLE III

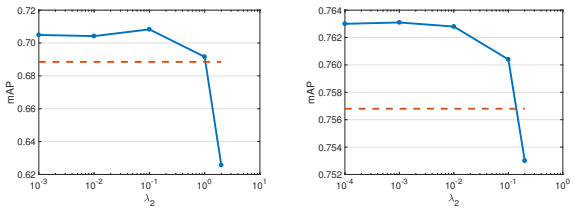
MAP COMPARISONS FOR VIDEO-TO-VIDEO RETRIEVAL TASK ON TWO DATABASES.

Method	YouTube Celebrities				UMDFaces			
	12-bit	24-bit	36-bit	48-bit	12-bit	24-bit	36-bit	48-bit
LSH [14]	0.1023	0.1360	0.2048	0.2078	0.0745	0.1360	0.2164	0.2690
SH [39]	0.1878	0.2289	0.2485	0.2490	0.1339	0.2070	0.2368	0.2563
ITQ [15]	0.3411	0.4751	0.5021	0.4990	0.2201	0.3652	0.4450	0.4746
SITQ [15]	0.3545	0.4521	0.5172	0.5745	0.2099	0.3639	0.4207	0.4587
BRE [21]	0.2646	0.2837	0.3590	0.3794	0.1689	0.2905	0.3396	0.3765
KSH [28]	0.4375	0.5427	0.6142	0.6442	0.2493	0.3738	0.4189	0.4522
DNNH [22]	0.5711	0.5722	0.6050	0.6743	0.4141	0.5620	0.6162	0.6213
DSH [26]	0.5148	0.5303	0.5578	0.5419	0.3418	0.5006	0.5599	0.6049
HashNet [6]	0.5005	0.6376	0.6655	0.6475	0.3391	0.4633	0.5431	0.5635
SSDH [42]	**0.5925**	0.6226	0.6451	0.6599	0.4656	0.6064	0.6456	0.6667
DVC [31]	0.5460	**0.6632**	0.6704	0.6820	**0.5656**	0.5837	0.6146	0.6204
HVIH	0.5602	0.6483	**0.6989**	**0.7042**	0.5570	**0.6846**	**0.7398**	**0.7628**

TABLE IV

MAP COMPARISONS FOR IMAGE-TO-VIDEO RETRIEVAL TASK ON TWO DATABASES.

Method	YouTube Celebrities				UMDFaces			
	12-bit	24-bit	36-bit	48-bit	12-bit	24-bit	36-bit	48-bit
SITQ [15]	0.3133	0.3944	0.4375	0.5005	0.1548	0.2790	0.3278	0.3469
KSH [28]	0.3885	0.4851	0.5410	0.5702	0.1882	0.2700	0.3070	0.3302
DNNH [22]	0.4670	0.4960	0.5267	0.5882	0.3160	0.3983	0.4242	0.4735
DSH [26]	0.4181	0.4753	0.5091	0.5005	0.2312	0.3378	0.3957	0.4412
HashNet [6]	0.4182	0.5558	0.5874	0.5774	0.2282	0.3240	0.3985	0.4270
SSDH [42]	**0.5468**	0.5784	0.6024	0.6120	0.3709	0.4876	0.5221	0.5464
HVIH	0.5159	**0.6023**	**0.6443**	**0.6569**	**0.4383**	**0.5428**	**0.5986**	**0.6237**



(a) (b)

Fig. 6. Retrieval performance variation tendencies on two datasets, i.e. (a) YouTube Celebrities (YTC) and (b) UMDFaces, when parameter λ_2 changes. Red dotted line denotes mAP results of baseline, i.e. setting without VCA module. The results are obtained with 48-bit binary codes.

V. CONCLUSIONS AND FUTURE WORKS

To address the problem of video face retrieval, we propose a compact and robust video representation learning framework. The superior retrieval performance of our method mainly deriveds from three aspects: Firstly, we retain frame-level feature in Hamming space and impose dense frame-wise supervision on them which improves the power of frames for subsequent video fusion. From the experimental results, we can see fully exploiting frame information enhances the performance remarkably. Secondly, we leverage metric learning technologies on video-level representation, which boosts the discriminability of video binary codes. Thirdly, our VCA module fine-tunes the position of videos, which results in better performance and more retrieval scenarios of learned codes (e.g. retrieval across image and video). For future work, three promising extensions would be investigated: 1) Compatibility with different video lengths in training phase, which can be handled by devising multi-similarity supervision manner to build more comprehensive and principled framework; 2) Exploitation of temporal context information with informative video modeling; 3) Application to the specific complicated event retrieval from massive surveillance videos.

VI. ACKNOWLEDGMENTS

This work is partially supported by Natural Science Foundation of China under contracts Nos. U19B2036, 61922080, 61772500, and CAS Frontier Science Key Research Project No. QYZDJ-SSWJSC009.

REFERENCES

[1] O. Arandjelovic and A. Zisserman. Automatic face recognition for film character retrieval in feature-length films. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 860–867, 2005.

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

[3] A. Ashraf, A. Yang, and B. Taati. Pain expression recognition using occluded faces. In *Proceedings of the IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*, pages 1–5, 2019.

[4] A. Bansal, A. Nanduri, C. D. Castillo, R. Ranjan, and R. Chellappa. Umdfaces: An annotated face dataset for training deep networks. In *IEEE International Joint Conference on Biometrics*, pages 464–473, 2017.

[5] M. Bauml, M. Tapaswi, and R. Stiefelhagen. Semi-supervised learning with constraints for person identification in multimedia data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3602–3609, 2013.

[6] Z. Cao, M. Long, J. Wang, and P. S. Yu. Hashnet: Deep learning to hash by continuation. In *Proceedings of the IEEE Conference on International Conference on Computer Vision*, pages 5608–5617, 2017.

[7] H. Cevikalp and B. Triggs. Face recognition based on image sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2567–2573, 2010.

[8] R. G. Cinbis, J. Verbeek, and C. Schmid. Unsupervised metric learning for face identification in tv video. In *Proceedings of the IEEE Conference on International Conference on Computer Vision*, pages 1559–1566, 2011.

[9] E. J. Crowley, O. M. Parkhi, and A. Zisserman. Face painting: querying art with photos. In *British Machine Vision Conference*, 2015.

[10] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[11] Z. Dong, S. Jia, T. Wu, and M. Pei. Face video retrieval via deep learning of binary hash representations. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[12] Z. Dong, C. Jing, M. Pei, and Y. Jia. Deep cnn based binary hash video representations for face retrieval. *Pattern Recognition*, 81:357–369, 2018.

[13] J. Feng, S. Karaman, and S.-F. Chang. Deep image set hashing. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pages 1241–1250, 2017.

[14] A. Gionis, P. Indyk, R. Motwani, et al. Similarity search in high dimensions via hashing. In *Proceedings of International Conference on Very Large Data Bases*, volume 99, pages 518–529, 1999.

[15] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2916–2929, 2012.

[16] M. He, J. Zhang, S. Shan, M. Kan, and X. Chen. Deformable face net: Learning pose invariant feature with pose aware feature alignment for face recognition. In *Proceedings of the IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*, pages 1–8, 2019.

[17] X. He, P. Wang, and J. Cheng. K-nearest neighbors hashing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2839–2848, June 2019.

[18] C. Jing, Z. Dong, M. Pei, and Y. Jia. Heterogeneous hashing network for face retrieval across image and video domains. *IEEE Transactions on Multimedia*, 21(3):782–794, 2018.

[19] M. Kim, S. Kumar, V. Pavlovic, and H. Rowley. Face tracking and recognition with visual constraints in real-world videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[20] T.-K. Kim, J. Kittler, and R. Cipolla. Discriminative learning and recognition of image set classes using canonical correlations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1005–1018, 2007.

[21] B. Kulis and T. Darrell. Learning to hash with binary reconstructive embeddings. In *Advances in Neural Information Processing Systems*, pages 1042–1050, 2009.

[22] H. Lai, Y. Pan, Y. Liu, and S. Yan. Simultaneous feature learning and hash coding with deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3270–3278, 2015.

[23] W. J. Li, S. Wang, and W. C. Kang. Feature learning based deep supervised hashing with pairwise labels. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2016.

[24] 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 *British Machine Vision Conference*, 2014.

[25] 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 *Proceedings of the IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*, volume 1, pages 1–8, 2015.

[26] H. Liu, R. Wang, S. Shan, and X. Chen. Deep supervised hashing for fast image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2064–2072, June 2016.

[27] W. Liu, R. Lin, Z. Liu, L. Liu, Z. Yu, B. Dai, and L. Song. Learning towards minimum hyperspherical energy. In *Advances in Neural Information Processing Systems*, pages 6222–6233, 2018.

[28] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang. Supervised hashing with kernels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2074–2081, 2012.

[29] Y. Liu, J. Yan, and W. Ouyang. Quality aware network for set to set recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5790–5799, 2017.

[30] T. Pfister, J. Charles, and A. Zisserman. Flowing convnets for human pose estimation in videos. In *Proceedings of the IEEE Conference on International Conference on Computer Vision*, pages 1913–1921, 2015.

[31] S. Qiao, R. Wang, S. Shan, and X. Chen. Deep video code for efficient face video retrieval. In *Asian Conference on Computer Vision*, pages 296–312. Springer, 2016.

[32] M. Raginsky and S. Lazebnik. Locality-sensitive binary codes from shift-invariant kernels. In *Advances in Neural Information Processing Systems*, pages 1509–1517, 2009.

[33] Y. Rao, J. Lin, J. Lu, and J. Zhou. Learning discriminative aggregation network for video-based face recognition. In *Proceedings of the IEEE Conference on International Conference on Computer Vision*, pages 3781–3790, 2017.

[34] C. Shan. Face recognition and retrieval in video. In *Video Search and Mining*, pages 235–260. Springer, 2010.

[35] J. Sivic, M. Everingham, and A. Zisserman. Person spotting: video shot retrieval for face sets. In *International Conference on Image and Video Retrieval*, pages 226–236. Springer, 2005.

[36] S. Su, C. Zhang, K. Han, and Y. Tian. Greedy hash: Towards fast optimization for accurate hash coding in cnn. In *Advances in Neural Information Processing Systems*, pages 806–815. 2018.

[37] J. Wang, S. Kumar, and S.-F. Chang. Semi-supervised hashing for scalable image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3424–3431, 2010.

[38] J. Wang, T. Zhang, N. Sebe, H. T. Shen, et al. A survey on learning to hash. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):769–790, 2017.

[39] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *Advances in Neural Information Processing Systems*, pages 1753–1760, 2009.

[40] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer, 2016.

[41] Z. Xu, Y. Yang, and A. G. Hauptmann. A discriminative cnn video representation for event detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1798–1807, 2015.

[42] H.-F. Yang, K. Lin, and C.-S. Chen. Supervised learning of semantics-preserving hash via deep convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(2):437–451, 2018.

[43] J. Yang, P. Ren, D. Zhang, D. Chen, F. Wen, H. Li, and G. Hua. Neural aggregation network for video face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4362–4371, 2017.

[44] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4694–4702, 2015.

[45] F. Zhao, Y. Huang, L. Wang, and T. Tan. Deep semantic ranking based hashing for multi-label image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1556–1564, 2015.

[46] W. Zheng, Z. Chen, J. Lu, and J. Zhou. Hardness-aware deep metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 72–81, 2019.

[47] Y. Zhong, R. Arandjelović, and A. Zisserman. Ghostvlad for set-based face recognition. In *Asian Conference on Computer Vision*, pages 35–50. Springer, 2018.