Unsupervised t-Distributed Video Hashing and its Deep Hashing Extension

DOI:
10.1109/TIP.2017.2737329

Document Version
Accepted author manuscript

Unsupervised t-Distributed Video Hashing and its Deep Hashing Extension

Yanbin Hao, Tingting Mu, *Member, IEEE,* John Y. Goulermas, *Senior Member, IEEE,* Jianguo Jiang, Richang Hong, *Member, IEEE,* and Meng Wang, *Member, IEEE,*

***Abstract*—In this work, a novel unsupervised hashing algorithm, referred to as t-USMVH, and its extension to unsupervised deep hashing, referred to as t-UDH, are proposed to support large-scale video-to-video retrieval. To improve robustness of the unsupervised learning, t-USMVH combines multiple types of feature representations and effectively fuses them by examining a continuous relevance score based on a Gaussian estimation over pairwise distances, and also a discrete neighbor score based on the cardinality of reciprocal neighbors. To reduce sensitivity to scale changes for mapping objects that are far apart from each other, Student t-distribution is used to estimate the similarity between the relaxed hash code vectors for keyframes. This results in more accurate preservation of the desired unsupervised similarity structure in the hash code space. By adapting the corresponding optimization objective and constructing the hash mapping function via a deep neural network, we develop a robust unsupervised training strategy for a deep hashing network. The efficiency and effectiveness of the proposed methods are evaluated on two public video collections via comparisons against multiple classical and state-of-the-art methods.**

***Index Terms*—Video retrieval, hashing, deep neural network, multi-view learning, unsupervised learning, Student t-distribution.**

I. Introduction

BOOSTED by the continuous development of internet technology and the popularity of digital products, video-related online activities, such as downloading, uploading, viewing and modifying, have gained significant increase of attention in the recent years [1]. This has resulted in a substantial amount of web video (or segment) data [2], and a high demand of content-based video retrieval that supports applications, such as near-duplicate video retrieval (NDVR) [3], [4], copy detection [5], video classification [6] and recommendation [7]. In general, a content-based video-to-video retrieval task follows a three-step procedure: (1) representation of a video by a sequence of frames, referred to as keyframes, extracted by uniform sampling [8] or shot-based methods [9]; (2) generation of video representation, e.g., content characterization of video keyframes (or segments); and (3) computation of similarities between the query video and the database videos based on the generated representations. Successful retrieval relies on the computation of a robust similarity score between videos. Therefore, it is essential to construct satisfactory video representations that characterize and quantify visual information in videos [10].

High retrieval accuracy is the principal priority of research in the field. Work on feature engineering aims at improving the retrieval performance by constructing high-quality video representations based on image/video domain knowledge. For instance, global signature features [11], [12] are advantageous for fast NDVR, but fail to represent longer and more complex videos than the near-duplicate ones. Optical flow [13] and dense trajectories [14] utilize local keypoints of keyframes to capture video motion information, and achieve good performance in action recognition. However, they are time-consuming approaches not capable of processing videos with complex scenes. To improve video representation, multi-view techniques have been developed. These capture video characteristics by mixing multiple feature types and analyzing connections from multiple perceptions [15]–[17]. In the recent years, deep learning has become the most effective technique for learning visual representations directly from pixels/voxels of images/videos. This paradigm offers substantial gain and performance improvement over traditional manual feature engineering in image/video classification and visual recognition [18]–[21]. Most state-of-the-art deep leaning systems are supervised, and although they offer excellent performance, they require tens of millions of labeled training instances. The performance of unsupervised deep learning systems is unfortunately not as successful as the supervised ones [21]. For example, the unsupervised training of a convolutional neural network (CNN) results in a significant performance drop compared to one with supervised training (e.g., AlexNet), both in terms of rate for the nearest neighbor retrieval task on the VOC 2012 dataset and classification accuracy for the scene classification task using the MIT Indoor set [21].

To support large-scale retrieval, another concern is the matching speed based on the learned video (or image) representation [1], [22], [23]. Currently, hashing is one of the most commonly used techniques that offers not only high retrieval speed, but also significantly reduced memory volume for storing videos (or images) [15], [24], [25]. Connecting it with representation learning, hashing can actually be deemed to be a binary representation learning approach that characterizes objects with binary codes.

To consider the issues of unsupervised deep learning and to facilitate large-scale retrieval, our work focuses on the

Y. Hao, J. Jiang, R. Hong and M. Wang are with the School of Computer and Information, Hefei University of Technology, Hefei, 230009, China (email: haoyanbin@hotmail.com, jgjiang@hfut.edu.cn, {hongrc.hfut, eric.mengwang}@gmail.com).

T. Mu is with School of Computer Science, University of Manchester, Kilburn Building, Oxford Road, Manchester, M13 9PL, UK (email: tingting.mu@me.com).

J. Y. Goulermas is with the Department of Computer Science, Ashton Building, University of Liverpool, Liverpool, L69 3BX, UK (email: j.y.goulermas@liverpool.ac.uk).

development of a robust unsupervised method for hash code learning, and its adaptation to a neural network to further improve unsupervised deep hashing. Building upon our previous work on multi-view hashing (SMVH) [26], we propose an unsupervised mechanism for constructing a composite similarity structure between keyframes supported by different types of feature representations. To capture the unsupervised similarity information more accurately, we take into account both the continuous relevance estimation based on a Gaussian distribution over pairwise distances and also the discrete neighbor relationships by examining the cardinality of the reciprocal neighbors. To preserve better the desired similarity structure in the hash code space, we propose to use Student t-distribution to estimate the similarity between the relaxed hash code vectors of the keyframes. This distribution imposes an inverse square law, which is beneficial with respect to large distances and noise effects from distant objects. By using a neural network to construct the hash mapping function, the model weight parameters can be optimized based on the composite Kullback-Leibler (KL) divergence computed between the desired unsupervised similarity structure and the computed hashing-based one. This provides a robust unsupervised training method for deep hashing models. Compared to the state-of-the-art hashing methods in [15], [26], the proposed one is also easier to optimize, since the utilization of the t-distribution in the construction of the probabilistic model simplifies the gradients of the cost function. The performance of the proposed method is demonstrated with benchmark evaluations and comparisons with state-of-the-art techniques.

The remaining of this paper is organized as follows. Section II briefly reviews related works. Section III outlines SMVH [26] which is the starting point of the current work. Section IV explains the proposed work on unsupervised similarity construction, t-distributed matching and the unsupervised deep hashing extension. In Section V the performance of the proposed methods is evaluated in terms of retrieval accuracy and efficiency, while Section VI concludes the presentation.

II. Related Work

A. Indexing Techniques

Video indexing studies mechanisms that represent video content in symbolic descriptions that allow search to be conducted by matching user queries. Sophisticated video indexing techniques have been developed to accelerate the search speed, such as tree-based [8], [27] and hashing [15], [26], [28]. Tree-based indexing partitions the video/image representation space from coarse to fine and forms a hierarchical tree structure [29]. One example work is [8] which presents a hierarchical filter-and-refine framework for video copy detection and copy segment localization. It first constructs a pattern-based index tree using symbol encoding as a filter, and then designs a pattern-based dynamic programming algorithm to re-rank the retrieved videos and to localize the copy segments. Another example work is [27] which proposes a two-level filtration approach using an adaptive vocabulary tree to index all the frame-level descriptors. Subsequently, it performs an edit-distance-based pairwise matching to detect video copies.

Hashing is used to encode an object (e.g., a video, an image, or a document) into a fixed-length binary string through a mapping strategy. Its advantages in retrieval include both fast distance computation and reduced memory costs. Example works include the locality sensitive hashing (LSH) [30], spectral hashing (SPH) [31] and the self-taught hashing (STH) [32]. These map a real-valued feature vector to a short binary string using random projections or binarizations of eigenvectors of the neighborhood graph constructed using object features. There are also non-spectral works, such as the supervised kNN hashing (kNNH) [33] and inductive manifold hashing (IMH) [34], [35] with both unsupervised and supervised versions. Their underlying optimizations are formulated via KL divergence. To improve the hashing performance, multi-view hashing techniques have also been developed to learn compact and efficient binary hash codes from a mixture of multiple feature views [15], [26], [36], [37]. Examples of unsupervised works relying on multi-view information, include the multiple feature hashing (MFH) [15], which learns hash codes for videos by manually weighting the importance of different types of feature sources, and also the multi-view alignment hashing (MAH) [36], which fuses the alignment representations from multiple sources while preserving the joint distributions. To enhance learning by incorporating label information, semi-supervised multi-view discrete hashing (SSMDH) [38] optimizes a composite objective function designed to serve multiple goals for pattern extraction.

Recently, there has been a boost in the development of deep hashing techniques. Deep neural networks (e.g., CNNs) are employed to learn binary representations for objects of interest through an appropriately selected activation function, such as a rectification linear or sign one [39]–[45]. Most of the state-of-the-art deep hashing approaches are CNN based, and can jointly generate feature representations and hash mappings. Training of these networks is usually supervised and relies on examples with known labels. They construct the training objective functions based on, for instance, pointwise labels [42], pairwise labels [43] and ranking labels [44], [45]. There are fewer deep learning works that generate hash codes in an unsupervised manner. Such an example, is deep hashing in [41], which optimizes the network weights by minimizing the quantization loss between the hash code returned by the output layer and the image representation from a hidden layer. Performance evaluation reports a significant drop of the unsupervised training compared to the supervised configuration.

B. Representation Learning

Video representations can be constructed by using hand-crafted visual features to characterize keyframes and by selectively combining time sequence information [13], [14], [46], [47]. One major feature extraction method [48], generates low-level features that characterize the global or local information of a given keyframe (or video). Global features (e.g., color features [49]) and extensions (e.g., color spatiograms [50] and Markov stationary features [51]) usually lead to fast retrievals [11], [12]. Local features rely on sets of local points and possess more superior discriminating power than the global

features, especially when characterizing objects with complex changes and scenes [52]. Popular local feature descriptors include SIFT [53], SURF [54], and LBP [55].

To improve the retrieval speed and accuracy, more sophisticated feature extraction methods have been developed. One example is [8], which groups local keypoint descriptors (e.g., SURF) into a fixed number of clusters using bag of words (BoW) [56], and assigns each cluster a unique "visual word". Subsequently, a keyframe (video clip) can be represented as a histogram of the occurrences of the visual word clusters, and the retrieval accuracy is closely related to the number of used clusters. Another example is multi-feature fusion (MFF) which exploits the complementary properties of the global and local features. Many MFF variations have been proposed to improve multimedia data representations [57]–[59]. In the recent years, deep neural networks are gradually replacing the conventional visual feature hand-crafting, because of their superior performance in multiple visual recognition challenges (e.g., CaffeNet, AlexNet [18] and R-CNN [60]). Although these networks can automatically learn excellent visual representations directly from image pixels, they require strong-supervision and their success relies on millions of manually labeled data, such as the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [19].

To reduce the dependency on labeled data, there has been a growing interest in developing unsupervised methods for video/image representation learning. For instance, [12], [61] conduct unsupervised representation learning by reinforcing the visual representations generated from hand-craft features through the use of freely available social tags or text descriptions of web videos. Neural networks are used to construct unsupervised feature representations via auto-encoders [62], [63] and restricted Boltzmann machines (RBMs) [64]. There are also works that use hand-crafted features (e.g., SIFT or HOG) to discover semantic classes [65], or to learn visual patches [21], and then employ the discovered classes and learned patches as the annotation information for the training.

III. Notations and Preliminaries

The proposed unsupervised hashing method is built upon our previous work on stochastic multi-view hashing (SMVH) [26] summarized with its relevant to this work notation as follows. Given a collection of V videos, a set of representative keyframes are firstly extracted for each video using either the uniform time sampling or shot-based sampling methods. Assume a total of n keyframes extracted from these V videos. For each keyframe, multi-view features are extracted to characterize its properties, where different feature views correspond to different types of feature representations (e.g., features extracted by different extraction methods). Assuming a total of m types of such representations, the gth feature type is stored in an $n \times d_g$ matrix $\mathbf{X}^{(g)} = [x_{ij}^{(g)}]$. Each column vector $\boldsymbol{x}_i^{(g)} = [x_{i1}^{(g)}, \ldots, x_{id_g}^{(g)}]^T$ denotes the gth feature vector for the ith keyframe. The SMVH model learns a set of s hash functions $\{\mathfrak{h}_i(\cdot)\}_{i=1}^{s}$, each taking the available features of a keyframe as the input and returning a binary number. Therefore, a total of s hash functions correspond to an s-length binary string for each keyframe. These computed strings are stored as the rows of the $n \times s$ binary hash code matrix $\mathbf{H} = [h_{ij}]$, with $\boldsymbol{h}_i = [h_{i1}, \ldots, h_{is}]^T$ where $h_{il} \in \{0, 1\}$ for $l = 1, \ldots, s$.

Given an ith keyframe characterized by multiple feature vectors $\{\boldsymbol{x}_i^{(g)}\}_{g=1}^{m}$, the hashing function is defined as

$$\mathfrak{h}_l\left(\left\{\boldsymbol{x}_i^{(g)}\right\}_{g=1}^{m}\right) = T\left(z_{il}\right), \tag{1}$$

where

$$z_{il} = \text{ sigmoid}\left(\sum_{g=1}^{m}\sum_{j=1}^{d_g} x_{ij}^{(g)} w_{lj}^{(g)} + b_l\right), \tag{2}$$

and $w_{lj}^{(g)}$ and b_l are the real weight parameters. The embedding vector $\boldsymbol{z}_i = [z_{i1}, \ldots, z_{is}]^T$ provides an approximation viewed as a relaxed version of the hash codes $\boldsymbol{h}_i$. The sigmoid function is used to convert the positive and negative embedding values to numbers close to one and zero. A thresholding function, defined as $T(x) = 1$, if $x > \theta$ and zero otherwise, is employed to binarize real-valued inputs.

Since the most critical component in a retrieval task is the video comparison guided by a similarity evaluation between videos, SMVH trains its hash code by preserving reliable similarity structure between the keyframes in the hash code space. This structure is the mixing of three elements. The first is the keyframe examination under the different views and its encoding by conditional probability matrices $\{\mathbf{P}^{(g)} = [p_{j|i}^{(g)}]\}_{g=1}^{m}$ estimated with Gaussian distributions as

$$p_{j|i}^{(g)} = \frac{\exp\left(-\frac{\left\|\boldsymbol{x}_i^{(g)} - \boldsymbol{x}_j^{(g)}\right\|_2^2}{2\sigma_{ig}^2}\right)}{\sum_{l \neq i}\exp\left(-\frac{\left\|\boldsymbol{x}_i^{(g)} - \boldsymbol{x}_l^{(g)}\right\|_2^2}{2\sigma_{ig}^2}\right)}. \tag{3}$$

The second is the within-video structure matrix $\mathbf{P}_W = [p_{ij}^{(W)}]$, with $p_{ij}^{(W)} = 1$ indicating that the ith and jth keyframes are from the same video and zero otherwise. The third and most important element, is the supervised label matrix $\mathbf{P}_S = [p_{ij}^{(S)}]$, with $p_{ij}^{(S)} = 1$ indicating that the ith and jth keyframes are extracted from matching videos and zero otherwise. To combine these, SMVH uses a soft voting scheme corresponding to the convex combination of these matrices

$$\mathbf{P} = N\left(\sum_{g=1}^{m}\alpha_g \mathbf{P}^{(g)} + \alpha_{m+1}\mathbf{P}_W + \alpha_{m+2}\mathbf{P}_S\right), \tag{4}$$

where $\{\alpha_i\}_{i=1}^{m+2}$ denote positive weights summing to one and $N(\cdot)$ normalizes each row of the input matrix.

The similarity structure preservation is realized by a matching procedure between the desired similarity matrix $\mathbf{P} = [p_{j|i}]$ and the computed similarity matrix, denoted by $\mathbf{Q} = [q_{j|i}]$, using hash codes. The matching score is examined by a

composite KL divergence to measure the difference between the two matrices $\mathbf{P}$ and $\mathbf{Q}$, given as

$$S_{KL} = \lambda \sum_{i=1}^{n} \sum_{j \neq i} p_{j|i} \log \frac{p_{j|i}}{q_{j|i}} + (1-\lambda) \sum_{i=1}^{n} \sum_{j \neq i} q_{j|i} \log \frac{q_{j|i}}{p_{j|i}}, \tag{5}$$

where λ is a user-defined parameter. The similarity $q_{j|i}$ is approximated from the relaxed hash code $\{\boldsymbol{z}_i\}_{i=1}^n$ of keyframes using Gaussian distribution, as

$$q_{j|i} = \frac{\exp\left(-\left\|\boldsymbol{z}_i - \boldsymbol{z}_j\right\|_2^2\right)}{\sum_{l \neq i} \exp\left(-\left\|\boldsymbol{z}_i - \boldsymbol{z}_l\right\|_2^2\right)}. \tag{6}$$

Finally, learning of the hash code is converted to the minimization of Eq.(5) with respect to the weights $w_{lj}^{(g)}$ and b_l. Using the relaxed hash codes $\{\boldsymbol{z}_i\}_{i=1}^n$, the video hash code can be finally generated by $h_{il}^{(v)} = T\left(\frac{1}{|\text{Ind}_i|} \sum_{j \in \text{Ind}_i} z_{jl}\right)$, where $h_{il}^{(v)}$ denotes the lth digit of the ith video's hash code, the set Ind_i denotes the keyframe indices of this video and $|\cdot|$ denotes set cardinality.

IV. Proposed Methods

To reduce the performance drop incurred by the lack of labeled information and build upon SMVH, we propose: (1) more accurate construction of an unsupervised similarity structure to be preserved by the hash code, and (2) more accurate preservation of the desired similarity in the hash code space. Additionally, motivated by the recent success of deep learning in computer vision, it is of practical interest to adapt the resulting robust unsupervised training strategy to a deep neural network architecture.

A. Unsupervised Similarity Construction

As pointed out in [26], a reliable similarity structure should be supported by the agreement between multiple types of feature representation. We proceed towards this direction and seek more accurate ways of encoding the multi-view similarity structure. Apart from the conditional probability matrices $\{\mathbf{P}^{(g)}\}_{g=1}^m$, discrete neighbor relationships are another important indicator that reveals a similarity structure between the keyframes; for example, being reciprocal neighbors indicates images are visually similar [66], [67]. Relying on this, we compute a relevance score based on reciprocal neighbors between objects under each view, given by a Jaccard coefficient

$$J_{ij}^{(g)} = \frac{|N_K^g(i) \cap N_K^g(j)|}{|N_K^g(i) \cup N_K^g(j)|}, \tag{7}$$

where $N_K^g(i)$ denotes the set of K nearest neighbors of the ith object searched under the gth feature view. This measure evaluates the percentage of the reciprocal neighbors among the existing neighbors of the two involved objects.

We further enrich this neighbor based relevance score with the conditional probability score as computed in Eq.(3), and accumulate the relevance over different types of features. We then obtain the following composite similarity score between two keyframes

$$p_{ij}^{(C)} = \frac{1}{2n} \sum_{g=1}^{m} J_{ij}^{(g)} \left(p_{j|i}^{(g)} + p_{i|j}^{(g)}\right). \tag{8}$$

To construct a symmetric similarity score, the conditional probability is symmetrized by $\frac{1}{2}(p_{j|i}^{(g)} + p_{i|j}^{(g)})$. The content-based video similarity matrix $\mathbf{P}_C = [p_{ij}^{(C)}]$ is then combined with the within-video structure matrix $\mathbf{P}_W$, to produce the final unsupervised similarity matrix

$$\mathbf{P} = (1-\alpha)\, N\left(\mathbf{P}_C\right) + \alpha N\left(\mathbf{P}_W\right), \tag{9}$$

where $N(\cdot)$ is a normalization function to restrict the input matrix to a sum of one. The parameter $0 < \alpha < 1$ balances the weights between the feature-driven content relevance and the video structure based relevance. The resulting matrix $\mathbf{P}$ is symmetric. Compared to the asymmetric one in Eq.(4), it greatly simplifies the gradients of the cost function and better circumvents the outlier problem [68]. Also, compared to Eq.(4), the proposed similarity construction employs fewer combination parameters, i.e., only one balancing parameter α compared to $m+2$. Both the proposed and SMVH employ an integer K to control the computation of the Gaussian width σ_{ig}; the same K also controls the neighbor based relevance score for the proposed method.

B. t-Distributed Structure Matching

Similar to Eq.(5), the KL divergence is used to match the similarity structures in the desired $\mathbf{P}$ from original videos and the computed $\mathbf{Q}$ from the hash codes. Given a fixed $\mathbf{P}$, the success of a good matching mainly relies on the construction of $\mathbf{Q}$. Inspired by the effectiveness of Student t-distribution in structure preservation for embedding computations [68], we employ it to estimate the relevance between keyframes based on their relaxed hash codes. Replacing accordingly the Gaussian distribution in Eq.(6) results in

$$q_{ij} = \frac{\left(1 + \left\|\boldsymbol{z}_i - \boldsymbol{z}_j\right\|_2^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \left\|\boldsymbol{z}_k - \boldsymbol{z}_l\right\|_2^2\right)^{-1}}. \tag{10}$$

The Student t-distribution can be viewed as an infinite mixture of Gaussians and bears the desirable property that $(1 + \|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2^2)^{-1}$ approaches an inverse square law for large pairwise distances $\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2$, which makes it almost invariant to scale changes for mapping objects far apart from each other.

Let $\boldsymbol{\theta} = \{\{w_{lj}^{(g)}\}_{l,j,g}, \{b_l\}_l\}$ be the set of variables used to parameterize the mapping function in Eq.(2) for computing the relaxed hash code. We use the newly defined p_{ij} described in Section IV-A to formulate the minimization problem

$$\min_{\boldsymbol{\theta}} O(\boldsymbol{\theta}) = S_{KL}\left(\boldsymbol{\theta}\right) + \mu R(\boldsymbol{\theta}). \tag{11}$$

The regularization term $R(\boldsymbol{\theta})$ is introduced to prevent overfitting, while $\mu > 0$ is the user-defined regularization parameter. In our case, one possibility for setting the regularization term is $R(\boldsymbol{\theta}) = \frac{1}{2} \sum_{g=1}^{m} \sum_{l=1}^{s} \sum_{j=1}^{d_g} (w_{lj}^{(g)})^2$. As the objective function is smooth and differentiable, we can employ stochastic

Algorithm 1 Pseudocode for the training of t-USMVH.

Input: n keyframes $\{\boldsymbol{x}_i^{(g)}\}_{i=1}^n$ extracted from V videos represented by m types of d_g-dimensional features ($g = 1,\ldots,m$).

Output: Combination coefficients $\{w_{l,j}^{(g)}\}$, bias parameters $\{b_l\}$.

Algorithmic parameters: Hash code length s, neighbor bound K, balancing parameter λ, regularization parameter μ, relevance matrix weight α.

Optimization parameters: Iteration number T, learning rate η and momentum $\zeta\,(t)$.

Initialization: Assign random values to the weight $\{w_{l,j}^{(g,0)}\}$ and bias variables $\{b_l^{(0)}\}$.

for $t = 1$ to T **do**

Compute gradient $\frac{\partial O}{\partial w_{lj}^{(g,t)}}$, compute gradient $\frac{\partial O}{\partial b_l^{(t)}}$.

Set the updates:

$w_{lj}^{(g,t+1)} = w_{lj}^{(g,t)} + \eta\frac{\partial O}{\partial w_{lj}^{(g,t)}} + \zeta(t)\left(w_{lj}^{(g,t)} - w_{lj}^{(g,t-1)}\right)$.

$b_l^{(t+1)} = b_l^{(t)} + \eta\frac{\partial O}{\partial b_l^{(t)}} + \zeta\,(t)\left(b_l^{(t)} - b_l^{(t-1)}\right)$.

end for

gradient descent (SGD) to find a reasonably good solution. The technical effort merely remains in the gradient computation with respect to $\boldsymbol{\theta}$, which we discuss in Section IV-D. The pseudocode of the training process is provided in Algorithm 1, where the parameters η and $\zeta\,(t)$ control the step size and T indicates the iteration number. The proposed method can be envisaged as an unsupervised extension of SMVH through Student t-distribution matching, and thus, we refer to it as t-USMVH. Its overall system structure is illustrated in Fig.1.

C. Deep Extension

In this section, we extend the previous unsupervised method to hash code generation through a neural network. The key idea is to use Eq.(5) as the loss function to train the network weights, where the computation of the unsupervised similarity structure to be preserved as well as the estimated similarity from the relaxed hash code follow the mechanisms described in Sections IV-A and IV-B. The main modification is that, instead of Eqs.(1,2), the relaxed hash code (embedding vector) for each keyframe is computed by a neural network (denoted as the function Φ) taking the whole keyframe image as input (denoted by I_i). This is given as

$$\begin{aligned} \mathfrak{h}_l(I_i) &= T\,(z_{il})\,, &(12)\\ [z_{i1}, z_{i2}, \ldots, z_{is}]^T &= \Phi(I_i, \boldsymbol{\theta}), &(13)\end{aligned}$$

where $\boldsymbol{\theta}$ is the set of weights to be optimized based on Eq.(11).

In this work, we use a CNN based on the typical LeNet-5 [69] to learn the relaxed hash code, with the input layer being fed with color images in RGB format. The structure of the proposed system is illustrated in Fig.2. Layer-wise based pre-training is applied. For learning each convolution/subsampling layer-pair, we fully connect a layer to the layer-pair and create a 3-layer CNN (not considering the input layer). Subsequently, we perform mini-batch SGD on the newly created CNN by using the proposed unsupervised training strategy. The sigmoid activation function is adopted in each convolution layer and the fully connected layer. After pre-training, the weights of the entire CNN are fine-tuned using the same objective function. We refer to this unsupervised deep hashing based on t-distribution matching as t-UDH. It generates image representations, that is binary thresholded hash codes, via learning from raw image pixels. To improve the accuracy of unsupervised learning, the learning procedure utilizes expert knowledge obtained by different feature extraction methods; this resembles multi-view learning. The use of the t-distribution improves the matching accuracy by reducing the noise effects of distant objects.

D. Gradient Computation

Here we provide the gradient computations for the introduced t-USMVH and t-UDH systems. Both models employ the same objective function in Eq.(11) for training, but for notational clarity, we decompose the objective function into the separate components

$$\begin{aligned} O &= \lambda\mathrm{KL}_1 + (1-\lambda)\mathrm{KL}_2 + \mu R, &(14)\\ \mathrm{KL}_1 &= \sum_{i=1}^{n}\sum_{t\neq i} p_{it}\log\frac{p_{it}}{q_{it}}, &(15)\\ \mathrm{KL}_2 &= \sum_{i=1}^{n}\sum_{t\neq i} q_{it}\log\frac{q_{it}}{p_{it}}. &(16)\end{aligned}$$

The objective function is controlled by the model variables $\boldsymbol{\theta} = [\theta_1,\ldots,\theta_{|\boldsymbol{\theta}|}]^T$, through the relaxed hash code (embedding) $\boldsymbol{z}_i = [z_{i1},\ldots,z_{is}]^T$. Applying chain rule gives

$$\frac{\partial O}{\theta_t} = \left[\lambda\frac{\partial \mathrm{KL}_1}{\partial z_{il}} + (1-\lambda)\frac{\partial \mathrm{KL}_2}{\partial z_{il}}\right]\frac{\partial z_{il}}{\partial \theta_t} + \mu\frac{\partial R}{\partial \theta_t}. \quad (17)$$

The target gradients $\{\frac{\partial O}{\theta_t}\}_{t=1}^{|\boldsymbol{\theta}|}$ depend on the different components $\frac{\partial \mathrm{KL}_1}{\partial z_{il}}$, $\frac{\partial \mathrm{KL}_2}{\partial z_{il}}$, $\frac{\partial z_{il}}{\partial \theta_t}$ and $\frac{\partial R}{\partial \theta_t}$. The differences of the gradient computation between t-USMVH and t-UDH lie in the computation of $\frac{\partial z_{il}}{\partial \theta_t}$. For t-USMVH, $\boldsymbol{\theta}$ includes $w_{lj}^{(g)}$ and b_l. As $z_{il} = \text{sigmoid}\,(\tilde{z}_{il})$ and $\tilde{z}_{il} = \sum_{g=1}^{m}\sum_{j=1}^{d_g} x_{ij}^{(g)} w_{lj}^{(g)} + b_l$, we can easily obtain that

$$\begin{aligned} \frac{\partial z_{il}}{\partial w_{lj}^{(g)}} &= \text{sigmoid}\,(\tilde{z}_{il})\,[1-\text{sigmoid}\,(\tilde{z}_{il})]\,x_{ij}^{(g)}, &(18)\\ \frac{\partial z_{il}}{\partial b_l} &= \text{sigmoid}\,(\tilde{z}_{il})\,[1-\text{sigmoid}\,(\tilde{z}_{il})]\,. &(19)\end{aligned}$$

For t-UDH, z_{il} corresponds to the image representation returned by the output layer and its gradient with respect to the network weights can be easily computed through backpropagation, which we do not discuss in detail. Computation of $\frac{\partial R}{\partial \theta_t}$ depends on the formulation of the used regularization term, which can be easily computed given a differentiable function. The remaining key computation to derive, which is shared by both t-USMVH and t-UDH, are

$$\frac{\partial \mathrm{KL}_1}{\partial \boldsymbol{z}_i} = \left[\frac{\partial \mathrm{KL}_1}{\partial z_{i1}}, \frac{\partial \mathrm{KL}_1}{\partial z_{i2}}, \ldots, \frac{\partial \mathrm{KL}_1}{\partial z_{il}}, \ldots, \frac{\partial \mathrm{KL}_1}{\partial z_{is}}\right]^T, \quad (20)$$

and

$$\frac{\partial \mathrm{KL}_2}{\partial \boldsymbol{z}_i} = \left[\frac{\partial \mathrm{KL}_2}{\partial z_{i1}}, \frac{\partial \mathrm{KL}_2}{\partial z_{i2}}, \ldots, \frac{\partial \mathrm{KL}_2}{\partial z_{il}}, \ldots, \frac{\partial \mathrm{KL}_2}{\partial z_{is}}\right]^T. \quad (21)$$

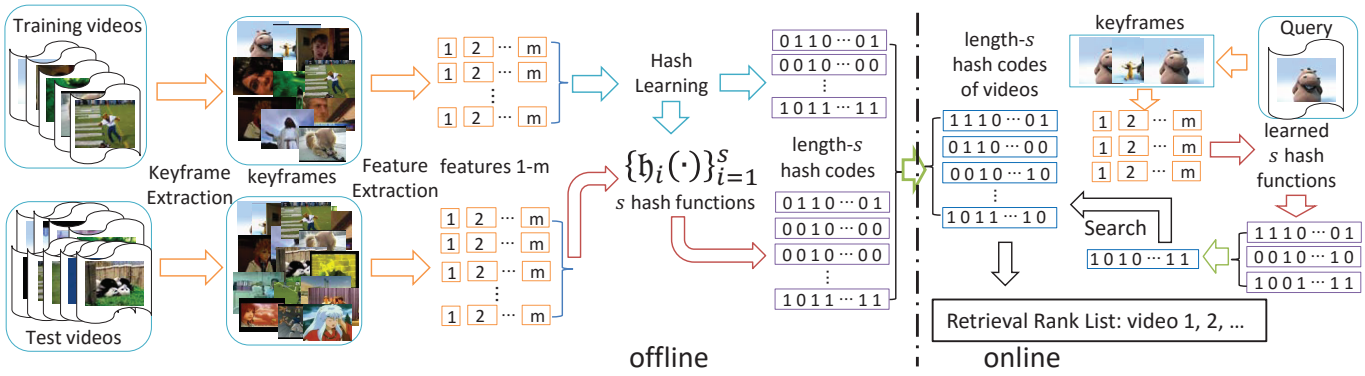


Fig. 1: Illustration of the overall architecture of the proposed video hashing system.

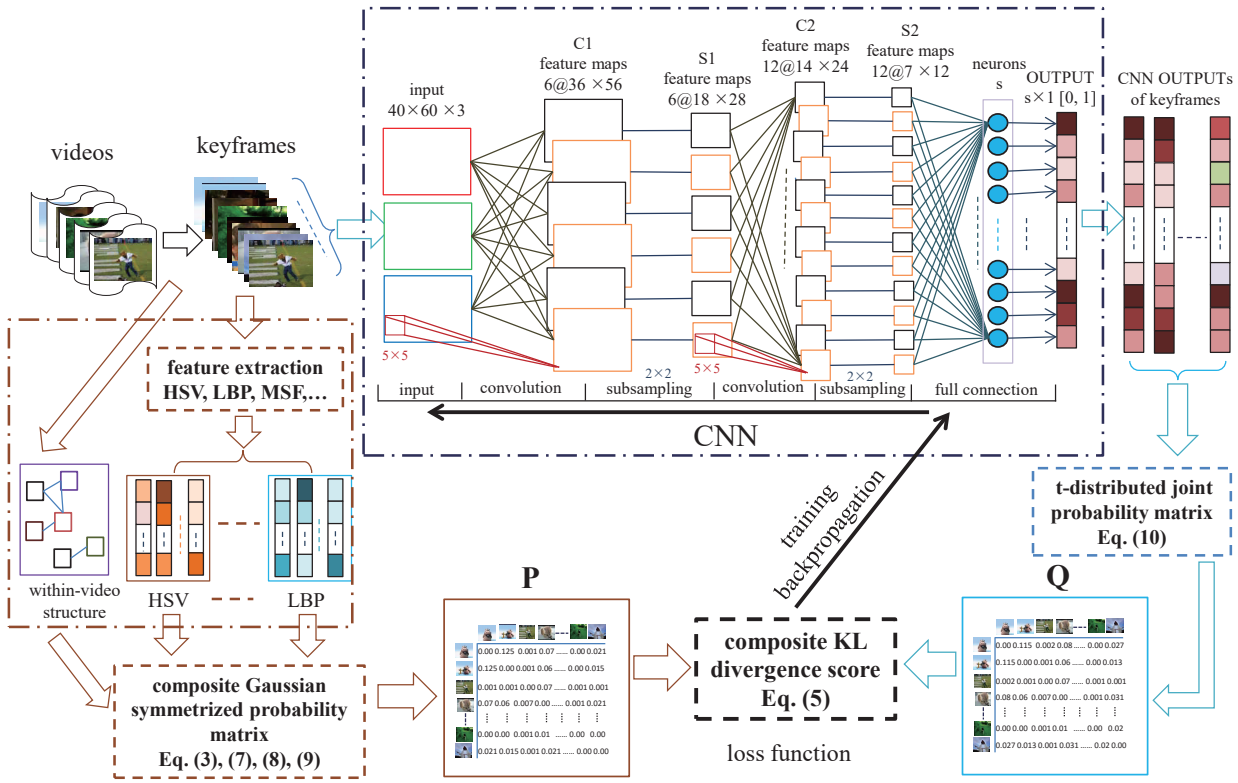


Fig. 2: Illustration of the overall architecture of the proposed unsupervised deep hashing system.

Following a similar procedure as in [68], we introduce the two auxiliary variables

$$d_{it} = \|\boldsymbol{z}_i - \boldsymbol{z}_t\|_2 , \tag{22}$$

$$U = \sum_{k \neq l} \left(1 + \|\boldsymbol{z}_k - \boldsymbol{z}_l\|_2^2\right)^{-1} = \sum_{k \neq l} \left(1 + d_{kl}^2\right)^{-1}, \tag{23}$$

to simplify the quantity

$$q_{it} = \frac{\left(1 + d_{it}^2\right)^{-1}}{U}. \tag{24}$$

Noting that $p_{it} = p_{ti}$ and $q_{it} = q_{ti}$ both in KL_1 and KL_2, and that d_{it} and d_{ti} possess exactly the same formulation, we have $\frac{\partial \mathrm{KL}_1}{\partial d_{it}} = \frac{\partial \mathrm{KL}_1}{\partial d_{ti}}$ and $\frac{\partial \mathrm{KL}_2}{\partial d_{it}} = \frac{\partial \mathrm{KL}_2}{\partial d_{ti}}$. Then, we have

$$\frac{\partial \mathrm{KL}_1}{\partial \boldsymbol{z}_i} = \frac{\partial \mathrm{KL}_1}{\partial d_{it}} \frac{\partial d_{it}}{\partial \boldsymbol{z}_i} + \frac{\partial \mathrm{KL}_1}{\partial d_{ti}} \frac{\partial d_{ti}}{\partial \boldsymbol{z}_i} = 2 \frac{\partial \mathrm{KL}_1}{\partial d_{it}} \frac{\partial d_{it}}{\partial \boldsymbol{z}_i}, \tag{25}$$

$$\frac{\partial \mathrm{KL}_2}{\partial \boldsymbol{z}_i} = \frac{\partial \mathrm{KL}_2}{\partial d_{it}} \frac{\partial d_{it}}{\partial \boldsymbol{z}_i} + \frac{\partial \mathrm{KL}_2}{\partial d_{ti}} \frac{\partial d_{ti}}{\partial \boldsymbol{z}_i} = 2 \frac{\partial \mathrm{KL}_2}{\partial d_{it}} \frac{\partial d_{it}}{\partial \boldsymbol{z}_i}. \tag{26}$$

We first derive $\frac{\partial \mathrm{KL}_1}{\partial d_{it}}$ and $\frac{\partial \mathrm{KL}_2}{\partial d_{it}}$ according to

$$\frac{\partial \mathrm{KL}_1}{\partial d_{it}} = \sum_{k \neq l} -p_{kl} \frac{\partial \left(\log q_{kl}\right)}{\partial d_{it}} = -\sum_{k \neq l} p_{kl} \frac{1}{q_{kl}} \frac{\partial q_{kl}}{\partial d_{it}}, \tag{27}$$

and

$$\frac{\partial \mathrm{KL}_2}{\partial d_{it}} = \sum_{k \neq l} \left[\frac{\partial \left(q_{kl} \log q_{kl} \right)}{\partial d_{it}} - \frac{\partial \left(q_{kl} \log p_{kl} \right)}{\partial d_{it}} \right] = \sum_{k \neq l} \left(1 + \log q_{kl} - \log p_{kl} \right) \frac{\partial q_{kl}}{\partial d_{it}}. \quad (28)$$

It can be seen that both derivatives in Eqs.(27,28) depend on $\frac{\partial q_{kl}}{\partial d_{it}}$, which can be calculated after incorporating Eq.(24) as

$$\frac{\partial q_{kl}}{\partial d_{it}} = \frac{1}{U} \frac{\partial \left(1 + d_{kl}^2\right)^{-1}}{\partial d_{it}} - q_{kl} \frac{1}{U} \frac{\partial U}{d_{it}}, \quad (29)$$

$$\frac{\partial U}{\partial d_{it}} = \sum_{k \neq l} \frac{\partial \left(1 + d_{kl}^2\right)^{-1}}{\partial d_{it}}. \quad (30)$$

Noting that $\frac{\partial \left(1+d_{kl}^2\right)^{-1}}{\partial d_{it}}$ is nonzero only when $k = i$ and $l = t$, and that $\sum_{k \neq l} p_{kl} = \sum_{k \neq l} q_{kl} = 1$, and we incorporate Eqs.(29,30) into Eqs.(27,28), we get

$$\frac{\partial \mathrm{KL}_1}{\partial d_{it}} = 2 p_{it} \left(1 + d_{it}^2\right)^{-1} d_{it} + \frac{1}{U} \frac{\partial U}{\partial d_{it}} = 2 \left(p_{it} - q_{it} \right) \left(1 + d_{it}^2\right)^{-1} d_{it}, \quad (31)$$

$$\frac{\partial \mathrm{KL}_2}{\partial d_{it}} = -2 \left(1 + \log \frac{q_{it}}{p_{it}} \right) \left(1 + d_{it}^2\right)^{-1} d_{it} + \sum_{k \neq l} \left(q_{kl} + q_{kl} \log \frac{q_{kl}}{p_{kl}} \right) \frac{1}{U} \frac{\partial U}{\partial d_{it}} = 2 \left(\sum_{k \neq l} q_{kl} \log \frac{q_{kl}}{p_{kl}} - \log \frac{q_{it}}{p_{it}} \right) q_{it} \left(1 + d_{it}^2\right)^{-1} d_{it}. \quad (32)$$

Subsequently, we have

$$\frac{\partial d_{it}}{\partial \boldsymbol{z}_i} = \frac{\partial d_{ti}}{\partial \boldsymbol{z}_i} = \frac{\boldsymbol{z}_i - \boldsymbol{z}_t}{d_{it}}. \quad (33)$$

Finally, substituting these into Eqs.(25,26), yields

$$\frac{\partial \mathrm{KL}_1}{\partial \boldsymbol{z}_i} = 4 \sum_t \left(p_{it} - q_{it} \right) \left(1 + \left\| \boldsymbol{z}_i - \boldsymbol{z}_t \right\|_2^2 \right)^{-1} \left(\boldsymbol{z}_i - \boldsymbol{z}_t \right), \quad (34)$$

$$\frac{\partial \mathrm{KL}_2}{\partial \boldsymbol{z}_i} = 4 \sum_t q_{it} \left(\sum_{k \neq l} q_{kl} \log \frac{q_{kl}}{p_{kl}} - \log \frac{q_{it}}{p_{it}} \right) \left(\boldsymbol{z}_i - \boldsymbol{z}_t \right) \left(1 + \left\| \boldsymbol{z}_i - \boldsymbol{z}_t \right\|_2^2 \right)^{-1}. \quad (35)$$

E. Discussion

The proposed t-USMVH is an unsupervised hashing algorithm formulated by minimizing the structural difference between the similarity matrices constructed in the original and embedded feature spaces. Compared to its predecessor SMVH [26] which is a supervised hashing method, the major challenge is how to improve the model design to limit the performance drop given the situation of lacking labeled examples. Compared to SMVH, the main changes we incorporate in the design of t-USMVH include a dedicated proximity calculation scheme between objects in the original space without involving any label information, and also an accurate structure preservation strategy utilizing Student t-distribution to estimate the embedded similarity structure and to reduce sensitivity to outlier objects. An additional benefit of t-USMVH is that it offers simpler gradient calculation than SMVH and this greatly facilitates the optimization procedure.

The previous work kNNH [33] is a supervised representative of formulating the hash code generation problem based on KL divergence. It utilizes the KL divergence to approximate kNN classification accuracy that is closely related to the intra-class neighbor retrieval precision. The IMH method [34], [35] employs t-SNE as its base algorithm to compute embeddings for anchor objects, and induces embeddings for query objects only from the anchor embeddings. During this process, the KL divergence approximates the neighbor retrieval precision for anchor objects. Differently from these methods, we take into account both neighbor retrieval precision and recall by constructing two KL-divergence scores. This forms a trade-off between precision and recall that can lead to a more accurate and balanced structure matching.

TABLE I: List of compared methods.

Acronym	Method description
VS	Retrieval by video signature [3].
HF	Retrieval by hierarchical filter [3].
SPH	Retrieval by spectral hashing [31].
STH	Retrieval by self-taught hashing [32].
IMH	Retrieval by inductive manifold hashing with t-SNE and 400 base samples [35].
MAH	Retrieval by multi-view alignment hashing [36].
MFH	Retrieval by multi-feature hashing [15].
USMVH	SMVH with $\alpha_{m+2} = 0$ [26].

V. Experimentation and Comparative Analysis

A. Experimental Setup

In the first experiment, we compare the proposed t-USMVH with the classical video-to-video retrieval systems including VS and HF, as well as the state-of-the-art ones based on various recent hashing techniques including SPH, STH, IMH, MAH, MFH and USMVH. All the compared methods are unsupervised, and their acronyms and descriptions are summarized in Table I. To evaluate the retrieval performance, the classic metric of the mean average precision (MAP) is employed. This is commonly used in the video retrieval community [3], [8], [15], [26]. Additionally, the precision-recall curve is provided to offer a more thorough view of the retrieval performance.

Two publicly available web video datasets are used to assess the retrieval performance. One is the CC_WEB_VIDEO dataset [3], which consists of 12,790 video clips downloaded from video sharing websites, such as YouTube, Google and Yahoo! via searching with different keywords, and subsequently organized into 24 sets. Within each set, the most popular video is used as the query, and the remaining videos are manually labeled by two non-expert assessors to create the ground truth. Shot boundaries of each video are detected and each shot is represented by a keyframe. There are a total of 398,015 keyframes extracted from this video collection. The other is the UQ_VIDEO dataset [15], which is a combined dataset

TABLE II: The parameter settings used for the t-USMVH.

Optimization param.	Value	Algorithmic param.	Value
T	1000	λ	0.9
η	500 (initially)	μ	0.01
$\zeta\,(t)\,(t<250)$	0.5	s	80-500
$\zeta\,(t)\,(t\geq 250)$	0.75	K	20

TABLE III: The tuning range of hash code length s.

Features	s_a	s_b	s_t
HSV (H)	100	160	20
MSF-color (M)	180	280	20
HSV, LBP (HL)	80	400	40
HSV, LBP, MSF-color (HLM)	100	500	100

by mixing the CC_WEB_VIDEO and the YouTube videos. There are a total of 169,952 videos and 2,570,554 keyframes extracted by a shot-boundary detection algorithm. Original keyframes and videos are available in CC_WEB_VIDEO, while UQ_VIDEO only provides features extracted by HSV (hue, saturation, value) and local binary patterns (LBP).

The proposed and the three competing methods MAH, MFH and USMVH are multi-view methods. They attempt to boost the retrieval performance by learning from different types of feature representations. Different feature extraction methods are used to construct different types of keyframe feature representations. In addition to the HSV features characterizing the global color histogram and LBP features characterizing the local texture, we also extract the color extension of the Markov stationary features (MSF) [27] for CC_WEB_VIDEO. The extracted MSF features not only characterize the spatial co-occurrence of histogram patterns but also incorporate local information. The extracted feature dimensionality is 162 for HSV, 256 for LBP and 288 for MSF.

The optimization parameters for t-USMVH follow the settings shown in the left side of Table II, and are determined from empirical recommendations from [68] for gradient descent updates. The algorithmic parameter settings are listed in the right side of Table II. The parameter η for controlling the step size is initially set to 500 and then updated in each iteration by means of the adaptive learning rate scheme described in [70]. The balancing parameter λ and the neighbor bound K are determined by following the empirical recommendations in [71]. The regularization parameter μ does not affect the performance much when it is within a reasonable range. The hash code length s is tuned from s_a to s_b with a step size s_t, as shown in Table III for different feature sets; these are set according to the input feature dimensionality. The within-video information weight α is tuned from 0.0 to 0.5. For the competing methods, we either use our implementation based on the settings recommended in their referenced work or employ the existing code provided by the authors. For the single-view methods SPH, STH and IMH, the features are included within a vector as their input. In all experiments, a random set of 600 videos are used for training. The online retrieval speed is recorded by Matlab R2012b on the same computer platform.

In the second experiment, we compare t-UDH with two commonly used unsupervised training strategies:

TABLE IV: The parameter settings used for the t-UDH.

Optimization parameter	Value
T_1	1500
T_2	1000
η_1	0.5
η_2	0.01
ζ	0.8

- One is based on an auto-encoder (AE) that minimizes the reconstruction error between the input and its estimation from the hidden layer representation. This is one of the most common unsupervised training methods.
- The other is based on the use of extra information resources from labeled corpora available for different but related visual tasks. Specifically, the two supervised CNN networks BVLC CaffeNet and BVLC R-CNN, which are trained on ILSVRC-2012 for image classification and on ILSVRC-2013 for object detection, are used to compress each video keyframe to a high-level representation vector of dimensionality 4,096. The obtained feature representations are referred to as CaffeNet_fc7 and R-CNN_fc7, and are used as the input to the t-USMVH system, but in its single view version ($g = 1$).

The CNN network trained by the proposed method and the AE are based on LeNet-5, in which there are two convolution layers, two subsampling layers and one full connection layer as shown in Fig.2. Secondly, we compare it with two state-of-the-art supervised deep hashing networks:

- The deep pointwise-supervised hashing (DSH) that trains a deep CNN to learn image representations and hash codes based on the pointwise training [42].
- The deep pairwise-supervised hashing (DPSH) that trains a deep CNN to learn image representations and hash codes based on the pairwise training [43].

For the proposed t-UDH training, the algorithm parameter settings are those in the right column of Table II, while the optimization parameters settings those in Table IV. The parameters T_1 and η_1 are the epoch number and learning ratio in each pre-training, while T_2 and η_2 are the corresponding parameters for the fine-tuning phase. The length of the hash code s corresponds to the neuron cardinality in the output layer of the CNN, and is set to vary from 200 to 500 with a step size of 100. The batch size (the number of training videos in each batch) for the SGD method is 100. Due to the fact that different videos may contain different numbers of keyframes, the number of keyframes in each batch between epochs may be different. A random set of 1,000 videos are used for training. The selected setting of $s = 400$ and $\alpha = 0.1$ is used by t-UDH to report the performance of the proposed training method (t-UDH) and the weakly supervised ones (CaffeNet_fc7 and R-CNN_fc7). The AE training of the CNN follows the layer-wise fashion proposed in [63]. For DSH and DPSH, we use the pre-trained CNN model CNN-F [72] to learn the image representation as recommended in the references. A total of 150 keyframes in each video set are randomly selected, resulting in $150 \times 24 = 3,600$ keyframes in the training data, and keyframes belonging to the same set are considered to

have the same label. The same hash code length of $s = 400$ is used for AE, DSH and DPSH training of the CNN.

B. Comparative Analysis for t-USMVH

For the proposed t-USMVH and its competing methods, we report their MAP performance computed with a total of 24 queries and their averaged online retrieval speed over the 24 searches for the two datasets in Tables V and VI, given different types of extracted features. The used parameter setting for reporting the performance in the second column of the tables is $s = 120$ and $\alpha = 0.3$ when the HSV (H) feature is used alone, $s = 220$ and $\alpha = 0.3$ for the MSF-color (M) feature alone, $s = 320$ and $\alpha = 0.25$ when both HSV and LBP (HL) are used, and $s = 400$ and $\alpha = 0.1$ for all HSV, LBP and MSF-color (HLM) features. We also compare the average precision-recall curves of these methods in Figs.3 and 4 for the two datasets. However, because HF is very time-consuming, we only examine its performance over the smaller dataset CC_WEB_VIDEO. It can be seen from the tables and the figures that the proposed t-USMVH outperforms all the competing methods under all the learning environments. For example, t-USMVH provides better performance than USMVH, and it always offers much better performance than VS, SPH, IMH given the same input features. It performs significantly better than MAH and MFH in many cases. It can also be seen that the MSF-color features provide in general better performance than the HSV features. This is mainly because the former not only characterize the spatial co-occurrence of histogram patterns, but also incorporate local information [27].

Fig.4(b) shows that for the large UQ_VIDEO dataset, the proposed system is able to achieve over 85% precision given recall values up to 75%. When it uses the HSV, LBP and MSF-color combination, we can find that the proposed t-USMVH gets the highest MAP value of 96.8% and also performs better than other methods on the CC_WEB_VIDEO dataset. Comparing with our former method USMVH which relies on Gaussian distributions to construct the similarity structures, these results also corroborate the advantage of the t-distribution. Regarding the retrieval speed, it can be seen from Tables V and VI that all the hashing-based systems are capable of achieving real-time retrievals even within the Matlab prototyping environment, whereas a longer hash code only leads to a slightly reduced retrieval speed.

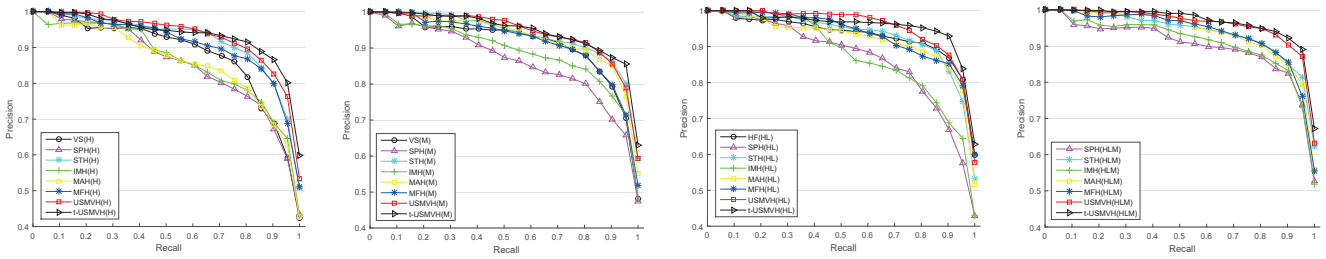
To examine how the proposed method performs when restricted to short hash code lengths, we conduct an evaluation by setting s to a small value ($s = 96$) and report the performance in the last columns of Tables V and VI for the two datasets. It can be seen that the proposed t-USMVH outperforms all the competing methods for all the compared feature views, which demonstrates its tolerance to short hash codes. We also investigate the behavior of t-USMVH given varying values of hash code lengths and the weight parameter α in Fig.5. It can be seen that t-USMVH performs well when $\alpha \in [0.1, 0.3]$. Also, when an effective value of α is selected, the algorithm is not very sensitive to the change of hash code lengths, e.g., the MAP performance is over 0.94 for most length values in $s \in [80, 400]$.

TABLE V: Performance comparisons with respect to MAP and retrieval speeds on the CC_WEB_VIDEO dataset. The best performances and speeds are boldfaced and second best underlined. "HP-S" denotes HSV and PCA-SIFT features.

Method	Features	MAP	Time (10^{-4} s)	MAP (s=96)
VS [3]	H	0.892	28.0	—
SPH [31]	H	0.854	4.91	0.857
STH [32]	H	0.922	4.95	0.922
IMH [35]	H	0.861	4.90	0.863
MAH [36]	H	0.859	4.97	0.849
MFH [15]	H	0.918	**4.81**	0.901
USMVH [26]	H	0.934	<u>4.82</u>	0.933
t-USMVH	H	0.937	**4.81**	0.937
VS [3]	M	0.913	38.4	—
SPH [31]	M	0.860	5.63	0.868
STH [32]	M	0.937	5.63	0.931
IMH [35]	M	0.888	5.62	0.870
MAH [36]	M	0.930	5.64	0.930
MFH [15]	M	0.929	5.60	0.905
USMVH [26]	M	0.943	5.61	0.934
t-USMVH	M	0.947	5.60	0.940
HF [3]	HP-S	0.952	>8000.0	—
HF [3]	HL	0.936	>8000.0	—
SPH [31]	HL	0.864	6.24	0.825
STH [32]	HL	0.932	6.50	0.918
IMH [35]	HL	0.875	6.30	0.835
MAH [36]	HL	0.921	6.41	0.842
MFH [15]	HL	0.928	6.38	0.898
USMVH [26]	HL	0.955	6.17	0.933
t-USMVH	HL	0.959	6.17	0.939
SPH [31]	HLM	0.901	6.94	0.871
STH [32]	HLM	0.940	7.11	0.935
IMH [35]	HLM	0.910	6.96	0.875
MAH [36]	HLM	0.939	7.02	0.918
MFH [15]	HLM	0.938	6.88	0.918
USMVH [26]	HLM	<u>0.962</u>	6.79	<u>0.947</u>
t-USMVH	HLM	**0.968**	6.79	**0.955**

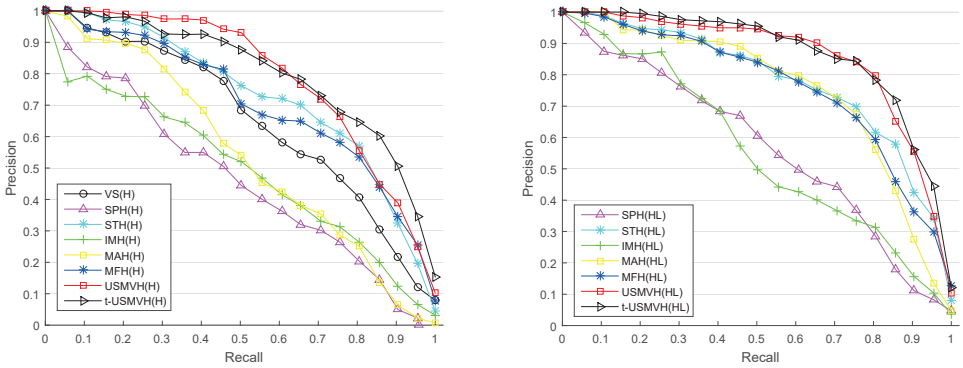
TABLE VI: Performance comparisons with respect to MAP and retrieval speeds on the UQ_VIDEO dataset. The best performances and speeds are boldfaced and second best underlined.

Method	Features	MAP	Time (s)	MAP (s=96)
VS [3]	H	0.640	0.2290	—
SPH [31]	H	0.457	0.0374	0.459
STH [32]	H	0.727	0.0386	0.712
IMH [35]	H	0.485	0.0377	0.464
MAH [36]	H	0.540	0.0391	0.545
MFH [15]	H	0.715	<u>0.0348</u>	0.680
USMVH [26]	H	0.787	**0.0345**	0.763
t-USMVH	H	0.792	**0.0345**	0.775
SPH [31]	HL	0.546	0.0671	0.510
STH [32]	HL	0.775	0.0682	0.704
IMH [35]	HL	0.550	0.0674	0.447
MAH [36]	HL	0.746	0.0680	0.540
MFH [15]	HL	0.757	0.0639	0.656
USMVH [26]	HL	<u>0.851</u>	0.0636	<u>0.768</u>
t-USMVH	HL	**0.858**	0.0637	**0.793**



(a) HSV (b) MSF-color (c) HSV and LBP (d) HSV, LBP and MSF-color

Fig. 3: Comparison of the average precision-recall curves for different methods computed using different features of the CC_WEB_VIDEO dataset. Parenthesized characters following the algorithm acronyms correspond to the used features.



(a) HSV (b) HSV and LBP

Fig. 4: Comparison of the average precision-recall curves for different methods computed using different features of the UQ_VIDEO dataset. Parenthesized characters are as in Fig.3.

To obtain a more detailed view of different methods, a bar graph comparing the average precision over each query is provided in Fig.6 using the UQ_VIDEO dataset. It can be seen that, for most queries, the multi-view version of t-USMVH performs better than its single-view version, although there also exist few individual cases such as Q5, Q11 and Q13 where the multi-view performance is not better. In some queries, such as Q7, Q11, Q13 and Q24, STH and MFH perform better than the t-USMVH. In general, when taking into account all the queries, the global and local features are complementary in achieving a more complete video representation, and it is therefore effective to combine both views. The proposed system provides the best overall retrieval performance.

C. Comparative Analysis for t-UDH

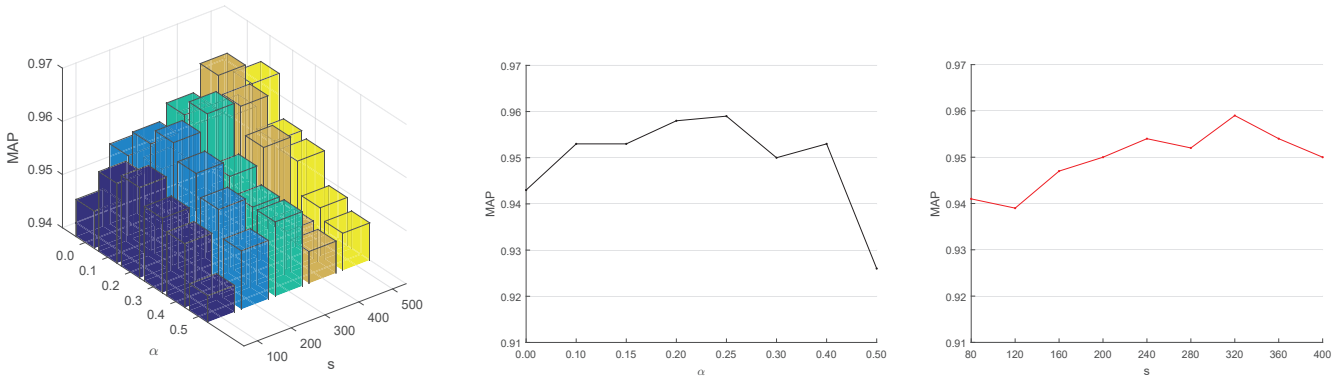
We compare the MAP performance of the CNN training for different methods using the CC_WEB_VIDEO dataset in Table VII, for which the corresponding precision-recall curves are plotted in Fig.7. The compared methods include: (1) the proposed t-UDH, where different mixtures of manual feature extraction methods are used to construct the desired similarity structure, (2) weakly supervised training using CaffeNet_fc7/R-CNN_fc7 features extracted from the "fc7" layer of two supervised CNN networks trained using two image corpora with labeled images, (3) AE, (4) AE / t-UDH using an AE for layer-wise pre-training and the proposed t-UDH for a fine-tuning of the entire network, and (5) the two supervised deep hashing methods DSH and DPSH based on pointwise and pairwise trainings, respectively.

Results show that DSH and DPSH set an upper bound in the retrieval performance. The t-UDH provides a satisfactory performance that is close to this upper bound when learning in a completely unsupervised manner. The unsupervised AE training provides the worst performance, which however, can be improved by employing the proposed t-UDH to fine-tune the network weights. The weakly supervised CNN using extra information provides good performance, but not the best. This is likely due to the fact that, although a large set of labeled images is used for training the CNN, the training and the NDVR data may contain different patterns.

As observed in previous experiments for t-USMVH (see Fig.5), when an effective value for α is used, the algorithm is not sensitive to the hash code length. Therefore, we fix the code length (the two cases of $s = 300$ and $s = 400$ are examined) and investigate the performance change of t-UDH by varying α in Fig.8 for two sets of feature views (HL and HLM). Similar to what is observed for t-USMVH, t-UDH performs in general well when its weight parameter is set within the range of $\alpha \in [0.1, 0.3]$.

VI. Conclusion

The novel unsupervised hashing algorithm t-USMVH and its extension to unsupervised deep hashing t-UDH were proposed to facilitate large-scale video-to-video retrieval. The



(a) HLM, $100 \leq s \leq 500, 0 \leq \alpha \leq 0.5$ (b) HL, $s = 320, 0 \leq \alpha \leq 0.5$ (c) HL, $\alpha = 0.25, 80 \leq s \leq 400$

Fig. 5: The MAP performance change of t-USMVH against different parameter values for the CC_WEB_VIDEO dataset with two different feature sets (HL and HLM).

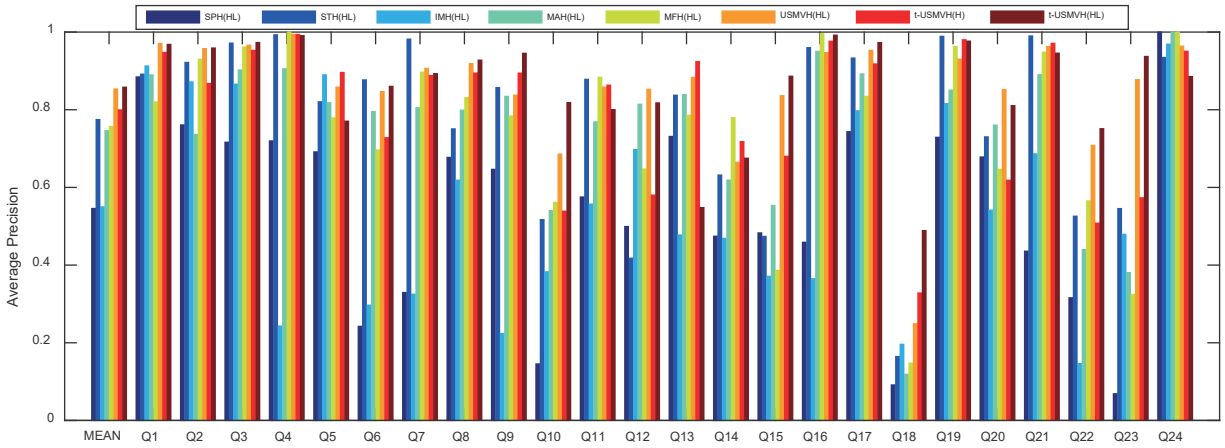


Fig. 6: Comparison of the average precision performance of different methods over each query (Q1-Q24) evaluated using the UQ_VIDEO dataset. The bar labeled "MEAN" represents the average MAP performance. Parenthesized characters follow the meaning used in Fig.3.

TABLE VII: MAP performance comparison of CNN network with different training strategies using CC_WEB_VIDEO data.

Methods	Type	Supporting Information	MAP
DSH [42]	Supervised	label information.	0.960
DPSH [43]	Supervised	label information	0.967
CaffeNet_fc7	Weakly supervised	labeled ILSVRC-2012 corpus for image classification	0.919
R-CNN_fc7	Weakly supervised	labeled ILSVRC-2013 corpus for object detection	0.926
AE	Unsupervised	NA	0.847
AE/t-UDH	Unsupervised	HSV and LBP feature extraction	0.919
t-UDH	Unsupervised	HSV and LBP feature extraction.	0.940
t-UDH	Unsupervised	HSV, LBP and MSF feature extraction.	0.957

first one addressed the accuracy, efficiency and scalability issues, as well as the lack of labeled images in training. These are all very important issues considered in recent video retrieval research and we contribute to their improvement. We achieved more accurate construction of between keyframe similarity without relying on label information and more accurate preservation of the desired similarity structure using hash codes with reduction of noise from distant keyframes. The proposed unsupervised method is robust and was further extended to train a deep neural network that improves the unsupervised deep hashing techniques. Results from extensive experimentations using public datasets showed the superior performance of the proposed methods over various classical and state-of-the-art algorithms.

ACKNOWLEDGEMENTS

This work was jointly supported by the "China Scholarship Council" and the "International Research Base for Developing Innovative Gerontechnology" sponsored by the national "111" Project (No. B14025) of China.

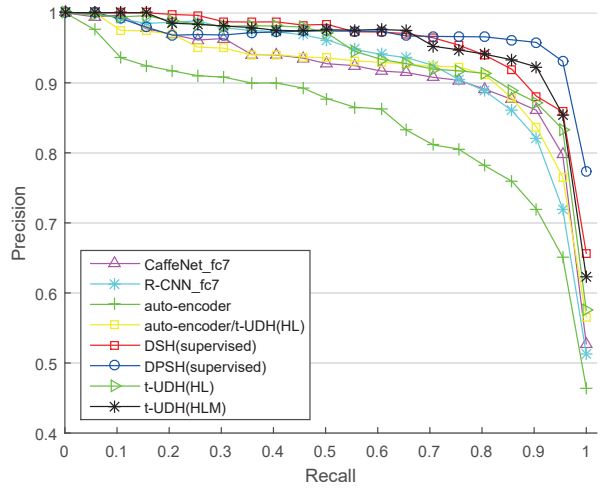
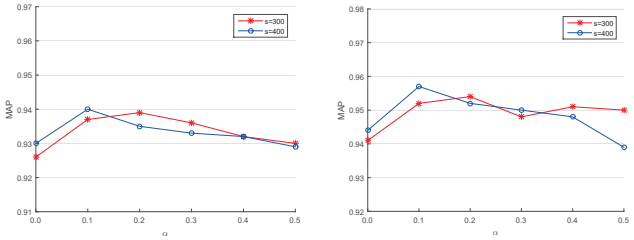


Fig. 7: Comparison of the average precision-recall curves for CNN with different training strategies on CC_WEB_VIDEO.



(a) HSV and LBP (b) HSV, LBP and MSF-color

Fig. 8: The MAP performance change of t-UDH against different parameter values for the CC_WEB_VIDEO dataset with two different feature sets.

REFERENCES

[1] J. Liu, Z. Huang, H. Cai, H. T. Shen, C. W. Ngo, and W. Wang, "Near-duplicate video retrieval: Current research and future trends," *ACM Computing Surveys (CSUR)*, vol. 45, no. 4, p. 44, 2013.

[2] M. Wang, R. Hong, G. Li, Z.-J. Zha, S. Yan, and T.-S. Chua, "Event driven web video summarization by tag localization and key-shot identification," *IEEE Trans. on Multimedia*, vol. 14, no. 4, pp. 975–985, 2012.

[3] X. Wu, A. G. Hauptmann, and C.-W. Ngo, "Practical elimination of near-duplicates from web video search," in *Proceedings of the 15th ACM international conference on Multimedia*, 2007, pp. 218–227.

[4] L. Shang, L. Yang, F. Wang, K.-P. Chan, and X.-S. Hua, "Real-time large scale near-duplicate web video retrieval," in *Proceedings of the 18th ACM international conference on Multimedia*, 2010, pp. 531–540.

[5] W.-L. Zhao and C.-W. Ngo, "Flip-invariant sift for copy and object detection," *IEEE Trans. on Image Processing*, vol. 22, no. 3, pp. 980–991, 2013.

[6] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[7] X. Zhou, L. Chen, Y. Zhang, L. Cao, G. Huang, and C. Wang, "Online video recommendation in sharing community," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, 2015, pp. 1645–1656.

[8] C.-L. Chou, H.-T. Chen, and S.-Y. Lee, "Pattern-based near-duplicate video retrieval and localization on web-scale videos," *IEEE Trans. on Multimedia*, vol. 17, no. 3, pp. 382–395, 2015.

[9] C. Cotsaces, N. Nikolaidis, and I. Pitas, "Video shot detection and condensed representation. a review," *IEEE signal processing magazine*, vol. 23, no. 2, pp. 28–37, 2006.

[10] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE trans. on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[11] Z. Huang, B. Hu, H. Cheng, H. T. Shen, H. Liu, and X. Zhou, "Mining near-duplicate graph for cluster-based reranking of web video search results," *ACM Trans. on Information Systems (TOIS)*, vol. 28, no. 4, p. 22, 2010.

[12] X. Wu, C.-W. Ngo, A. G. Hauptmann, and H.-K. Tan, "Real-time near-duplicate elimination for web video search with content and context," *IEEE Trans. on Multimedia*, vol. 11, no. 2, pp. 196–207, 2009.

[13] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," *Image analysis*, pp. 363–370, 2003.

[14] H. Wang, D. Oneata, J. Verbeek, and C. Schmid, "A robust and efficient video representation for action recognition," *International Journal of Computer Vision*, vol. 119, no. 3, pp. 219–238, 2016.

[15] J. Song, Y. Yang, Z. Huang, H. T. Shen, and J. Luo, "Effective multiple feature hashing for large-scale near-duplicate video retrieval," *IEEE Trans. on Multimedia*, vol. 15, no. 8, pp. 1997–2008, 2013.

[16] J. Chen, Y. He, and J. Wang, "Multi-feature fusion based fast video flame detection," *Building and Environment*, vol. 45, no. 5, pp. 1113–1122, 2010.

[17] S. Sun, "A survey of multi-view machine learning," *Neural Computing and Applications*, vol. 23, no. 7-8, pp. 2031–2038, 2013.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[20] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in *International Conference on Machine Learning*, 2015, pp. 843–852.

[21] X. Wang and A. Gupta, "Unsupervised learning of visual representations using videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2794–2802.

[22] M. Chen, S. Mao, and Y. Liu, "Big data: a survey," *Mobile Networks and Applications*, vol. 19, no. 2, pp. 171–209, 2014.

[23] J. Song, Y. Yang, Z. Huang, H. T. Shen, and R. Hong, "Multiple feature hashing for real-time large scale near-duplicate video retrieval," in *Proceedings of the 19th ACM international conference on Multimedia*, 2011, pp. 423–432.

[24] Z. Liu, H. Li, W. Zhou, R. Zhao, and Q. Tian, "Contextual hashing for large-scale image search," *IEEE Trans. on Image Processing*, vol. 23, no. 4, pp. 1606–1614, 2014.

[25] J. Wang, H. T. Shen, J. Song, and J. Ji, "Hashing for similarity search: A survey," *arXiv.org*, 2014. [Online]. Available: http://arxiv.org/abs/1408.2927

[26] Y. Hao, T. Mu, R. Hong, M. Wang, N. An, and J. Y. Goulermas, "Stochastic multiview hashing for large-scale near-duplicate video retrieval," *IEEE Trans. on Multimedia*, vol. 19, no. 1, pp. 1–14, 2017.

[27] M.-C. Yeh and K.-T. Cheng, "Video copy detection by fast sequence matching," in *Proceedings of the ACM International Conference on Image and Video Retrieval*, 2009, p. 45.

[28] M. Li and V. Monga, "Robust video hashing via multilinear subspace projections," *IEEE Trans. on Image Processing*, vol. 21, no. 10, pp. 4397–4409, 2012.

[29] L.-f. Ai, J.-q. Yu, Y.-f. He, and T. Guan, "High-dimensional indexing technologies for large scale content-based image retrieval: a review," *Journal of Zhejiang University SCIENCE C*, vol. 14, no. 7, pp. 505–520, 2013.

[30] A. Gionis, P. Indyk, R. Motwani *et al.*, "Similarity search in high dimensions via hashing," in *VLDB*, vol. 99, no. 6, 1999, pp. 518–529.

[31] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Advances in neural information processing systems*, 2009, pp. 1753–1760.

[32] D. Zhang, J. Wang, D. Cai, and J. Lu, "Self-taught hashing for fast similarity search," in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, 2010, pp. 18–25.

[33] K. Ding, C. Huo, B. Fan, and C. Pan, "knn hashing with factorized neighborhood representation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1098–1106.

[34] F. Shen, C. Shen, Q. Shi, A. Van Den Hengel, and Z. Tang, "Inductive hashing on manifolds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1562–1569.

[35] F. Shen, C. Shen, Q. Shi, A. van den Hengel, Z. Tang, and H. T. Shen, "Hashing on nonlinear manifolds," *IEEE Trans. on Image Processing*, vol. 24, no. 6, pp. 1839–1851, 2015.

[36] L. Liu, M. Yu, and L. Shao, "Multiview alignment hashing for efficient image search," *IEEE Trans. on Image Processing*, vol. 24, no. 3, pp. 956–966, 2015.

[37] L. Zheng, S. Wang, and Q. Tian, "Coupled binary embedding for large-scale image retrieval," *IEEE Trans. on Image Processing*, vol. 23, no. 8, pp. 3368–3380, 2014.

[38] C. Zhang and W.-S. Zheng, "Semi-supervised multi-view discrete hashing for fast image search," *IEEE Trans. on Image Processing*, vol. 26, no. 6, pp. 2604–2617, 2017.

[39] W.-J. Li, S. Wang, and W.-C. Kang, "Feature learning based deep supervised hashing with pairwise labels," *arXiv.org*, 2015. [Online]. Available: http://arxiv.org/abs/1511.03855

[40] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, "Supervised hashing for image retrieval via image representation learning," in *AAAI*, vol. 1, 2014, pp. 2156–2162.

[41] V. Erin Liong, J. Lu, G. Wang, P. Moulin, and J. Zhou, "Deep hashing for compact binary codes learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2475–2483.

[42] K. Lin, H.-F. Yang, J.-H. Hsiao, and C.-S. Chen, "Deep learning of binary hash codes for fast image retrieval," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 27–35.

[43] W.-J. Li, S. Wang, and W.-C. Kang, "Feature learning based deep supervised hashing with pairwise labels," *arXiv.org*, 2015. [Online]. Available: http://arxiv.org/abs/1511.03855

[44] R. Zhang, L. Lin, R. Zhang, W. Zuo, and L. Zhang, "Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification," *IEEE Trans. on Image Processing*, vol. 24, no. 12, pp. 4766–4779, 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*, 2015, pp. 1556–1564.

[46] J. Chang, D. Wei, and J. W. Fisher, "A video representation using temporal superpixels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2051–2058.

[47] B. Solmaz, S. M. Assari, and M. Shah, "Classifying web videos using a global video descriptor," *Machine vision and applications*, vol. 24, no. 7, pp. 1473–1485, 2013.

[48] Y. Li, "Video representation," in *Encyclopedia of Database Systems*, 2009, pp. 3296–3303.

[49] B. S. Manjunath, J.-R. Ohm, V. V. Vasudevan, and A. Yamada, "Color and texture descriptors," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 703–715, 2001.

[50] S. T. Birchfield and S. Rangarajan, "Spatiograms versus histograms for region-based tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2005, pp. 1158–1163.

[51] J. Li, W. Wu, T. Wang, and Y. Zhang, "One step beyond histograms: Image representation using markov stationary features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[52] W. Dong, Z. Wang, M. Charikar, and K. Li, "Efficiently matching sets of features with random histograms," in *Proceedings of the 16th ACM international conference on Multimedia*, 2008, pp. 179–188.

[53] D.-G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the 7th IEEE international conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157.

[54] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer vision–ECCV 2006*, 2006, pp. 404–417.

[55] G. Zhao and M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 915–928, 2007.

[56] N. Passalis and A. Tefas, "Entropy optimized feature-based bag-of-words representation for information retrieval," *IEEE Trans. on Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1664–1677, 2016.

[57] Y. Yang, J. Song, Z. Huang, Z. Ma, N. Sebe, and A. G. Hauptmann, "Multi-feature fusion via hierarchical regression for multimedia analysis," *IEEE Trans. on Multimedia*, vol. 15, no. 3, pp. 572–581, 2013.

[58] V. Ranjan, N. Rasiwasia, and C. Jawahar, "Multi-label cross-modal retrieval," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4094–4102.

[59] D. Wang, X. Gao, X. Wang, L. He, and B. Yuan, "Multimodal discriminative binary embedding for large-scale cross-modal retrieval," *IEEE Trans. on Image Processing*, vol. 25, no. 10, pp. 4540–4554, 2016.

[60] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[61] M. Mazloom, X. Li, and C. Snoek, "Tagbook: A semantic video representation without supervision for event detection," *IEEE Trans. on Multimedia*, vol. 18, no. 7, pp. 1378–1388, 2016.

[62] Q. V. Le, "Building high-level features using large scale unsupervised learning," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013, pp. 8595–8598.

[63] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *International Conference on Artificial Neural Networks*, 2011, pp. 52–59.

[64] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 609–616.

[65] B. C. Russell, W. T. Freeman, A. A. Efros, J. Sivic, and A. Zisserman, "Using multiple segmentations to discover objects and their extent in image collections," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 1605–1614.

[66] H. Jegou, C. Schmid, H. Harzallah, and J. Verbeek, "Accurate image search using the contextual dissimilarity measure," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 2–11, 2010.

[67] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas, "Query specific rank fusion for image retrieval," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 37, no. 4, pp. 803–815, 2015.

[68] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 2579-2605, p. 85, 2008.

[69] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[70] R. A. Jacobs, "Increased rates of convergence through learning rate adaptation," *Neural networks*, vol. 1, no. 4, pp. 295–307, 1988.

[71] J. Venna, J. Peltonen, K. Nybo, H. Aidos, and S. Kaski, "Information retrieval perspective to nonlinear dimensionality reduction for data visualization," *Journal of Machine Learning Research*, vol. 11, no. Feb, pp. 451–490, 2010.

[72] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *arXiv.org*, 2014. [Online]. Available: http://arxiv.org/abs/1405.3531

Yanbin Hao is a Ph.D. student in School of Computer and Information, Hefei University of Technology. His research interests mainly include machine learning and multimedia data analysis, such as large-scale multimedia indexing and retrieval, and multimedia data embedding.

Tingting Mu received the B.Eng. degree in electronic engineering and information science from the University of Science and Technology of China, Hefei, China, in 2004, and the Ph.D. degree in electrical engineering and electronics from the University of Liverpool in 2008. She is currently a Lecturer in the Department of Computer Science at the University of Manchester. Her current research interests include machine learning, data visualization and mathematical modeling, with applications to information retrieval, text mining, and bioinformatics.

John Y. Goulermas (M'98, S'10) obtained the B.Sc.(1st class) degree in computation from the University of Manchester (UMIST), in 1994, and the M.Sc. and Ph.D. degrees from the Control Systems Center, UMIST, in 1996 and 2000, respectively. He is currently a Reader in the Department of Computer Science at the University of Liverpool. His current research interests include machine learning, combinatorial data analysis, data visualization as well as mathematical modeling. He has worked with various application areas including image/video analysis, biomedical engineering and biomechanics, industrial monitoring and control, and security. He is a senior member of the IEEE and an Associate Editor of the IEEE Transactions on Neural Networks and Learning Systems.

Jianguo Jiang is a professor with the School of Computer Science and Information, Hefei University of Technology, China. He is also a special expert who enjoys the special subsidy of the State Council and also a deputy director of Anhui Institute of Computing Science. Since 1985 he has been engaged in digital image analysis/processing and computer-aided monitoring system, and either presided over or participated in as many research projects as over 30. He has been awarded several Prizes for scientific progress, including: 1) one Second Class Prize and one Third Class Prize of National Scientific Progress of China; 2) one Second Class Prize of Ministry of Coalmining Industry of China; and 3) one First Class Prize and one Second Class Prize of Anhui Scientific Progress.

Richang Hong received the Ph.D. degree from the University of Science and Technology of China, Hefei, China, in 2008. He was a Research Fellow with the School of Computing, National University of Singapore, Singapore, from September 2008 to December 2010. He is currently a Professor with the Hefei University of Technology. His current research interests include multimedia question answering, video content analysis, and pattern recognition. He has co-authored more than 50 publications in his areas of expertise. Dr. Hong is a member of the Association for Computing Machinery (ACM). He was the recipient of the Best Paper Award in ACM Multimedia 2010.

Meng Wang is a professor at the Hefei University of Technology, China. He received his B.E. and Ph.D. degrees in the Special Class for the Gifted Young and the Department of Electronic Engineering and Information Science from the University of Science and Technology of China (USTC), Hefei, China, respectively. His current research interests include multimedia content analysis, search, mining, recommendation, and large-scale computing. He received the best paper awards successively from the 17th and 18th ACM International Conference on Multimedia, the best paper award from the 16th International Multimedia Modeling Conference, the best paper award from the 4th International Conference on Internet Multimedia Computing and Service, and the best demo award from the 20th ACM International Conference on Multimedia.