Active Video Hashing via Structure Information Learning for Activity Analysis

XIANGDONG WANG, QIUXU WANG, AND HUI WANG
School of Physical Education, Shandong Normal University, Jinan 250014, China
Corresponding author: Xiangdong Wang (xiangdong.wang.sdn@outlook.com)

This work was supported by the Project of National Social Science Foundation Study on the Influence of Sedentary Behavior on College Students’ Healthy Constitution and Intervention Method under Grant 15BTY055.

ABSTRACT
When attempting to analyze and understand large-scale video datasets, choosing training videos using active learning can significantly reduce the annotation costs associated with supervised learning without sacrificing the accuracy of classifiers. However, to further reduce the computational overhead of exhaustive comparisons between low-level visual feature with high dimensions, we have developed a novel video hashing coding model based on an active learning framework. The model optimizes mean average precision in a straightforward way by explicitly considering the structure information in video clips when learning the optimal hash functions. The structure information considered includes the temporal consistency between successive frames and the local visual patterns shared by videos with same semantic labels. Rather than relying on the similarity between paired videos, we use a ranking-based loss function to directly optimize mean average precision. Then, combined with the active learning component, we jointly evaluate the unlabeled training videos according to the uncertainty and average precision values with an efficient algorithm based on structured SVM. Extensive experimental results over several benchmark datasets demonstrate that our approach produces significantly higher search accuracy than traditional query refinement schemes.

INDEX TERMS
Active learning, video analysis.

I. INTRODUCTION
Due to the popularity of digital cameras and the emergence of social media websites, such as YouTube, Facebook and Youku, the amount of videos that can be accessed by everyone has increased exponentially over the past decade. Facing such proliferation of videos, how to efficiently analyze and understand the contents of videos in large scale has become a significant challenge in the field of computer vision [1]–[3]. In machine learning, training videos with high quality annotations is definitely beneficial to better performance. However, it is usually tedious and time consuming to label videos in large scale; especially, the accurate and reliable annotations for some specific area should be given by some experts possessed of the corresponding background knowledge. These issues severely limit the application of supervised learning in practice, where a large amount of videos are collected without labels.

In the past decades, although some studies have attempted to address the issues above through semi-supervised learning, to the best of our knowledge, these approaches typically suppose all unlabeled training videos are equally informative and input all of them at once [4]. In fact, on one hand, substantial unlabeled data might cause large uncertainties and potential threats [5], [6]; On the other hand, an arbitrary unlabeled training videos may even be redundant [7]. Active learning technique, as a complementary way in supervised learning, adaptively distinguish the informative videos for which labels should be requested. As a result, this technique reduce the annotation cost and simultaneously do not sacrifice the accuracy. Recent studies have suggested the effectiveness and efficiency of active learning in many areas, such as machine learning, data mining, information retrieval, computer vision [8]. The challenge of active learning for video understanding is how to decide whether a video is “informative” or “useful” [9]–[14]. From this perspective, there are two criteria in the literature for selecting training videos to be annotated, i.e., representativeness sampling and
uncertainty sampling. For example, Tong and Chang [15] proposed the support vector machine (SVM) active learning algorithm to conduct effective relevance feedback concept accurately and quickly. In this algorithm the informative training data is defined as the ones close to the classification boundary of SVM classifier. In a later work, Wang et al. [16] incorporate the unlabeled training data into the learning procedure of bootstrapping. Besides the uncertainty information, Brinker [17] incorporated a diversity measure into the active learning.

Another significant difficulty in large scale video analysis stems from the high-dimension low-level visual features representation of videos with complex contents, which usually suffers from the curse of dimensionality. Besides of the semantic gap between low-level features and high-level concepts, it is computationally costly and practically prohibitive for content-based retrieval through exhaustive comparisons of low-level visual feature, when handling a large collection of video clips [18]–[20]. To address this issue, video hashing has attracted considerable attention from researchers and practitioners because of its specific requirements, robustness, security and computational efficiency over millions or even billions of video clips [21]. In the past decades, a variety of hashing techniques have been exploited [22]. Some researchers propose to encode videos into a set of compact hash codes through incorporating various machine learning techniques while preserving the underlying geometric structure of the original data, such as locality sensitive hashing (LSH) [23], self-taught hashing [24], spectral hashing [25], and so on. However, some existing video hashing approaches directly employ still image-hashing techniques for each frame of video and then concatenate these frame-level hashing code into a high-dimensional binary vector [26], [27]. This strategy costs computationally for the high-dimensional hash code as well as the extensive frames of long videos [28]. Although some studies select only one key frame to represent the videos [29], it is sensitive to frame dropping and noise [28]. In fact, one video consists of multiple frames with dynamic backgrounds and evolving people activities [30], and thus the temporal information plays a significant role in the content understanding of videos. Making use of this information is beneficial to achieve better performance.

To take the temporal evolution of video content information into consideration, some efforts are devoted to exploiting the correlations between the sequence of frames. For example, in the field of robust video hashing, Coskun et al. [31] were based on low-pass coefficients of 3D cosine-based transforms and represent a video segment as a three-dimensional matrix; Esmaeili and Ward [28] divided the sequence into video segments, and then generate a temporally informative representative image of each segment by linearly combining the frames in this segment. In the field of content-based video retrieval, Ye et al. [18] explicitly took into account the specific structure of structure information in video clips. This method achieves promising results over some benchmark datasets; however, it learns the optimal hashing functions based on the similarity (distance) between pairs of training videos but do not optimize the ultimate retrieval performance straightforwardly. Additionally, the hash coding method proposed in [18], [32] are conducted in the framework of supervised learning, which might not be desirable for many applications where a large number of videos are collected without labels.

In this paper, we are based on the framework of active learning and develop a novel video hashing coding model to optimize the average precision (AP) straightforwardly. This model explicitly take the structure information of video clips into consideration, including the temporal consistency between successive frames together with the local visual patterns shared by videos with same semantic labels. We learn the hash functions based on ranking-based loss function that maximize mean average precision (MAP) performance. Note that the order of the relevant videos are preserved in the framework of AP optimization rather than the purely absolute similarity between each pair of videos. Instead of using supervised learning, we involve the proposed video hash coding with structure information into a framework of active learning to mitigate the tedious work of manual annotation for large scale videos. Specifically, our algorithm starts with a small number of seed labeled videos, followed by linear hash functions learning with structure information of video clips. Based on the current hash functions, some training videos are selected to be annotated by evaluating the uncertainty and AP of the candidate videos in the active pool, where the criterion related to AP ensure the winner videos selected for labeling increase the performance of AP.

The contributions of this new model lie in three aspects

1) Considering the structure information of video clips, we learn the linear hash functions by optimizing the average precision (AP) straightforwardly.

2) We involve the proposed video hash coding into the framework of active learning, where unlabeled video candidates are evaluated according to the uncertainty and the value of AP.

3) An efficient algorithm is exploited to solve the proposed challenging problems based on structured SVM. Extensive experimental results demonstrate the effectiveness and efficiency of this algorithm.

II. AP OPTIMIZATION WITH VIDEO HASHING

In this section, we first introduce the frame-level hash coding based on linear embedding and define the similarity measurement of videos in binary subspace. Then, we are based on the structured SVM and develop a general framework of AP optimization with video hashing.

A. HASH CODING OF VIDEO

We suppose that the training set consisting of $N$ videos is denoted by $X = \{X^i \in \mathbb{R}^{n \times n_i} : i = 1, 2, \cdots, N\}$, where the $k$-th video $X^i = \{x_1^i, x_2^i, \cdots, x_n^i\} \in \mathbb{R}^{d \times n_i}$ contains $n_i$ successive frames $x_t^i \in \mathbb{R}^d$ for $t = 1, 2, \cdots, n_i$ and
Given any video frame \( x_i \), we focus on learning its \( K \)-bit binary code \( h(x_i) = [h_1(x_i), h_2(x_i), \ldots, h_K(x_i)]^T \in \mathbb{R}^K \), i.e., design \( K \) binary hash functions as linear hashing function \( h_k : \mathbb{R}^d \to \{-1, 1\}^K \),
\[
h_k(x_i) = \text{sign}(w_k^T x_i + b_k)
\] (1)
for \( k = 1, 2, \ldots, K \), where \( w_k \in \mathbb{R}^d \) refers to a hashing (classification/decision) hyperplane for the \( k \)-th bit of the code. For a better representation, we collect all \( w_k(k = 1, 2, \ldots, K) \) into matrix \( W = [w_1, w_2, \ldots, w_K] \in \mathbb{R}^{d \times K} \), namely transformation matrix for hashing coding. \( b_k \) is the intercept which equals to 0 if the frames in training dataset are zero mean. Sign function \( \text{sign}(\cdot) \) (element-wise) is used in this paper for its simplicity and efficiency.

Following an effective metric methodology in [33]–[35], we define the similarity of hash codes between query video \( Q \in \mathcal{Q} \) and videos in training set \( X \) through function \( S: \mathcal{Q} \times \mathcal{X} \to \mathbb{R} \), for video query \( Q = [q_1, q_2, \ldots, q_n] \in \mathcal{Q} \) and the \( i \)-th video \( X^i = [x_{i1}, x_{i2}, \ldots, x_{in}] \in \mathcal{X} \),
\[
S(Q, X^i) = \frac{1}{n_q n_h} \sum_{a=1}^{n_q} \sum_{b=1}^{n_h} s(q_a, x_{ib})
\] (2)
where the function \( s : \mathbb{R}^d \times \mathbb{R}^d \to [-1, 1] \) is defined to measure the cosine-similarity between video frames by
\[
s(q_a, x_{ib}) = \frac{h(q_a)^T h(x_{ib})}{\|h(q_a)\| \cdot \|h(x_{ib})\|} = \frac{1}{K} h(q_a)^T h(x_{ib}).
\] (3)
This definition suggests that the cosine-similarity of hash codes between a pair of videos equals to the average cosine-similarity of each pair of frames. It encourages most of frames involved in a video play significant roles in measuring the similarity between videos.

However, the similarity measurement defined above is not tractable because the discrete nature of sign function in each linear hash function makes it non-differentiable and hard to optimize. Intuitively, we relax the Eq. (2) by replacing the sign function with its signed magnitude. In such a way, similar videos not only share same sign but also large projection magnitudes, meanwhile dissimilar videos have not only different signs but also as far as possible. Therefore, the Eq. (2) can be reformulated as
\[
S(Q, X^i) = \frac{1}{Kn_q n_h} \sum_{a=1}^{n_q} \sum_{b=1}^{n_h} (W^T q_a)^T (W^T x_{ib})
\] (4)
where \( \langle \cdot, \cdot \rangle_F \) denotes the Frobenius inner products and scatter matrix \( S(Q, X^i) \in \mathbb{R}^{n_q \times d} \) is defined as
\[
S(Q, X^i) = \frac{1}{Kn_q n_h} \sum_{a=1}^{n_q} \sum_{b=1}^{n_h} q_a (x_{ib})^T
\] (5)
Note that the descending order of \( S(Q, X^i) \) \( (i = 1, 2, \cdots, N) \) based on the cosine-similarity is equivalent to the ascending order of Hamming distance between the hash codes of video query \( Q \) and videos in training set \( \mathcal{X} \). In fact, the cosine-similarity has been proved to generate the identical result with one based on Hamming distance [33].

### B. AP Optimization Based on Hash Coding of Video

Given video query \( Q \in \mathcal{Q} \), we denote by \( \mathcal{X}^+_Q \subseteq \mathcal{X} \) and \( \mathcal{X}^-_Q \subseteq \mathcal{X} \) the subsets of relevant and irrelevant videos in training dataset \( \mathcal{X} \). It is said that video \( X^i \) is ranked higher than \( X^j \) if \( y_{ij} = 1 \) \((y_{ij} = -1)\) in a corresponding ranking matrix \( Y = [y_{ij}]_{N \times N} \in \mathcal{Y} \), \( \mathcal{Y} \subseteq \mathbb{R}^{N \times N} \) refers to the set of rankings (permutations) of training dataset \( \mathcal{X} \).

Following the framework of structural SVM, the optimal ranking \( Y \in \mathcal{Y} \) based on hash codes can be obtained by solving the following problem
\[
\Psi(Q, Y) = \sum_{i \in \mathcal{X}^+_Q, j \in \mathcal{X}^-_Q} y_{ij} \left( \frac{|S(Q, X^i) - S(Q, X^j)|}{|X^+_Q| - |X^-_Q|} \right) \] (6)
where \( \mathcal{K}(Q, Y) = \sum_{i \in \mathcal{X}^+_Q, j \in \mathcal{X}^-_Q} y_{ij} \frac{|S(Q, X^i) - S(Q, X^j)|}{|X^+_Q| - |X^-_Q|} \) is the ground truth ranking of videos in training set \( \mathcal{X} \) given video query \( Q \) is specified in this paper as \( y^*_{ij} = 1 \) \((y^*_{ij} = -1)\) if video \( X^i \in \mathcal{X}^+_Q \) and video \( X^j \in \mathcal{X}^-_Q \) for any \( i, j = 1, 2, \ldots, N \).

Based on the partial order feature representation, we define a discriminant function \( \Psi : \mathcal{Q} \times \mathcal{Y} \to \mathbb{R} \) by
\[
\Psi(Q, Y) = \sum_{i \in \mathcal{X}^+_Q, j \in \mathcal{X}^-_Q} y_{ij} \left( \frac{|S(Q, X^i) - S(Q, X^j)|}{|X^+_Q| - |X^-_Q|} \right)
\] (7)
As a result, the optimal ranking \( Y \) of training videos \( X^i \in \mathcal{X} \) \( (i = 1, 2, \cdots, N) \) given video query \( Q \) can be achieved simply in descending order of \( S(Q, X^i) \) \( (i = 1, 2, \cdots, N) \) when the hash functions \( h_k \) \( (k = 1, 2, \cdots, K) \) are fixed. This prediction rule is exactly satisfied since the descending order of \( S(Q, X^i) \) \( (i = 1, 2, \cdots, N) \) is equivalent to the ascending order of the Hamming distance between hash codes of query video \( Q \) and video \( X^i \) \( (i = 1, 2, \cdots, N) \).

Following the framework of structural SVM, the ranking problem given video query \( Q \in \mathcal{Q} \) is cast to learn linear hash function \( h_k \) \( (k = 1, 2, \cdots, K) \) which are parameterized by hashing hyperplane parameters matrix \( W \in \mathbb{R}^{d \times K} \). Specifically, given each video query \( Q \in \mathcal{Q} \), we firstly define the disagreement between any ranking \( Y \in \mathcal{Y} \) and the ground truth ranking \( Y^* \) based on discriminant function \( \Psi \) by \( \delta_{\Psi} : \mathcal{Q} \times \mathcal{Y} \to \mathbb{R} \),
\[
\delta_{\Psi}(Q, Y^*, Y) = \Psi(Q, Y^*) - \Psi(Q, Y)
\] (8)
Intuitively, function \( \delta_{\Psi}(Q, Y^*, Y) \) measures how compatible the current predicted ranking \( Y \) is with the grand truth ranking...
Y* give video query Q ∈ Q. The structure SVM is applied to maximize the margins between the ground truth ranking Y* and all the other possible rankings Y ∈ Y given each video query Q ∈ Q, i.e., for any Q ∈ Q and Y ∈ Y, the following inequality holds

\[ \delta_Y(Q, Y^*, Y) \geq \nabla(Y^*, Y) - \xi Q \]  

(9)

where \( \xi Q \geq 0 \) is a slack variable and the empirical ranking loss with average precision is defined as

\[ \nabla(Y^*, Y) = 1 - AP(Y^*, Y) \]  

(10)

and the average precision (AP) \( AP(Y^*, Y) \) is formulized by

\[ AP(Y^*, Y) = \frac{1}{|X_q^+|} \sum_{k=1}^{|X_q^+|} Prec@k(Y) \mathbf{1}_{X_k \in X_q^+} \]  

(11)

where \( Prec@k(Y) \) denotes the percentage of relevant videos in the top k videos in predicted ranking \( Y \); \( \mathbf{1}_{X_k \in X_q^+} \) is an indicator function which is defined as \( \mathbf{1}_{X_k \in X_q^+} = 1 \) if \( X_k \in X_q^+ \) and \( \mathbf{1}_{X_k \in X_q^+} = 0 \), otherwise.

### III. THE PROPOSED METHODOLOGY

In this section, we concern the structure information of videos and encode this information into the similarity measurement of video hashing codes. Following such strategy, we propose a novel video hash coding model in the framework of AP optimization and exploit an efficient algorithm to solve the challenging problem.

#### A. VIDEO HASHING WITH STRUCTURE INFORMATION

Recall that we represent each video as finite number of successive frames given a video-level label, and thus it is imperative to constrain that the two hash code of successive frames are close to each other as much as possible [18]. Given hashing transformation matrix \( W \), we concatenate all successive frame pair differences of video training set \( \mathcal{X} \) into matrix \( \bar{X} \in \mathbb{R}^{d \times T} \) whose columns are defined as \( \bar{X}_i = X_{i+1} - X_i \) for \( i = 1, 2, \ldots, N; t = 1, 2, \ldots, n_t - 1 \), where \( T = \sum_{t=1}^{N} (n_t - 1) \). Then the overall difference between each pair of successive frames in training set \( \mathcal{X} \) is formulized as a function of the hash transformation matrix \( W \), i.e.,

\[ T(W) = \sum_{j=1}^T \| (W^T \bar{X})_{:,j} \|_\infty \]  

(12)

where \( (W^T \bar{X})_{:,j} \) refers to the \( j \)-th column of matrix \( W^T \bar{X} \) for \( j = 1, 2, \ldots, T \); The \( \ell_\infty \)-norm of any vector \( z = [z_1, z_2, \ldots, z_d]^T \in \mathbb{R}^d \) is defined as

\[ \| z \|_\infty = \max_{1 \leq i \leq d} |z_i| \]  

(13)

Intuitively, minimizing the overall difference \( T(W) \) based on \( \ell_\infty \)-norm indeed ensures stronger constraint such that the hash codes of each successive frames are close to each other.

Based on the structure information of the videos in training set \( \mathcal{X} \), we learn the optimal hashing hyperplane parameters matrix \( W \in \mathbb{R}^{d \times K} \) in the framework of structured SVM by solving the following optimization problem,

\[ \min_{W, \xi \geq 0, \nu} \frac{1}{N} \sum_{i=1}^N \xi_i + \lambda \| W \|_{2,1} + \gamma T(W) \]

\[ \text{s.t.} \forall i = 1, 2, \ldots, N; \forall Y \in \mathcal{Y}, \]

\[ \delta_Y(X_i, Y_i^*, Y) \geq \nabla(Y_i^*, Y) - \xi_i \]  

(14)

where \( \lambda > 0 \) and \( \gamma > 0 \) are trade-off parameters; \( \xi_i \geq 0 \) \( (i = 1, 2, \ldots, N) \) are slack variables; \( Y_i^* \in \mathcal{Y} \) refers to the ground truth ranking of videos in training set \( \mathcal{X} \) given the i-th video query \( X_i \in \mathcal{X} \) for \( i = 1, 2, \ldots, N \). The \( \ell_{2,1} \)-norm of matrix \( W = [w_{ij}] \in \mathbb{R}^{d \times K} \) is defind as

\[ \| W \|_{2,1} = \sum_{i=1}^d \sqrt{\sum_{j=1}^K w_{ij}^2} = \sum_{i=1}^d \| W_{i, -} \|_2 \]  

(15)

where \( W_{i, -} \) refers to the i-th row of matrix \( W \). Thus, the \( \ell_{2,1} \)-norm based regularizer with trade-off parameter \( \lambda \) aims to learn an efficient hash transformation matrix \( W \) with only a small number of rows are non-zero. In such a way, some specific dimensions of frame feature that corresponds to the zero rows of \( W \) are capable of being ignored for a better robustness [36]; At the same time, this strategy makes the learned video frame features lie in a local common patterns related to a certain categories and convey more discriminative information [18].

For efficiency and compactness, we follow the idea use in [37] and employ only a single slack variable across all the constraints in optimization problem (14), namely the “1-slack” formulation. This strategy leads to less computation time than the traditional structural SVM [37], [38]. Thus, the optimization problem (14) for optimal solution of hash transformation matrix \( W \) is reformulated as

\[ \min_{W, \xi \geq 0} \xi + \lambda \| W \|_{2,1} + \gamma T(W) \]

\[ \text{s.t.} \forall (Y_1, Y_2, \ldots, Y_N) \in \mathcal{Y}^N, \]

\[ \frac{1}{N} \sum_{i=1}^N \delta_Y(X_i, Y_i^*, Y_i) \geq \frac{1}{N} \sum_{i=1}^N \nabla(Y_i^*, Y_i) - \xi \]  

(16)

where \( Y_i \in \mathcal{Y} \) for \( i = 1, 2, \ldots, N \) and \( (Y_1, Y_2, \ldots, Y_N) \in \mathcal{Y}^N \) collects a batch of rankings corresponding to the N videos in training set \( \mathcal{X} \). It has been proved in [37] that the formulations of problems (14) and (16) are equivalent to each other.

#### IV. JOINT ACTIVE HASH LEARNING SCHEME

In this section, we follows the framework of pool-based active learning [39], [40] and select new videos only from the pool of unlabeled videos. The most important issue lies in how to measure the informativeness of the unlabeled videos. We develop an approach to actively select a set \( \mathcal{X}_k = \{X_k : k = 1, 2, \ldots, L\} \) consisting of L informative videos for user to label, namely the active set of videos. With the procedure of active selection, on one hand, we can avoid labeling millions of videos in large scale dataset; on the other
hand, the more informative videos can be utilized to learn a better hashing function. In this paper, two criteria are considered to measure the informativeness of video, including the certainty of the unlabeled video and estimation of average precision.

A. CRITERION RELATED TO CERTAINTY
Recall that in the hashing model of video with structure information, the binary hashing code of video frame $x_i^t$, $h(x_i^t) = (h_1(x_i^t), h_2(x_i^t), \ldots, h_K(x_i^t))^\top \in \mathbb{R}^K$ is learned via thresholding $K$ linear hashing functions $h_k(x_i^t)$ ($k = 1, 2, \ldots, K$) parameterized with (classification/decision) hyperplane matrix $W = (w_1, w_2, \ldots, w_K) \in \mathbb{R}^{d \times K}$, i.e., $h(x) = \text{sign}(W^\top x)$. As a result, each hyperplane $w_k$ corresponds to one bit of the hashing code. The $k$-th hashing bit of video frame $x_i^t$ is 1, i.e., $h_k(x_i^t) = 1$ if $x$ sits on the positive side of a decision hyperplane $w_k$, and $h_k(x_i^t) = -1$ otherwise. As a result, the certainty of video $X$ is defined on the basis of its structure information by:

$$A_1(X) = \|W^\top X\|_F$$

where $\| \cdot \|_F$ represents the Frobenius norm of a matrix. Intuitively, the criterion formulated as Eq. (17) ensures means that the smaller the distance of a video $X$ to the hyperplane matrix $W$, the more certain the video is. Considering an extreme case where all of frames involved in one video lies exactly on the hyperplanes, then it is definitely certain to decide whether its corresponding bit should be 1 or -1. In other words, the smaller the value of certainty, the more informative of the video.

According to this criterion, the total certainty of $L$ selected videos is formulated as $\sum_{x \in \mathcal{X}_A} \|W^\top X\|_F$, and therefore, we can select a set of videos that are most informative by solving the following optimization problem:

$$\min_{\mathcal{X}_A} \sum_{x \in \mathcal{X}_A} A_1(X) \quad \text{s.t.} \quad |\mathcal{X}_A| = L$$

B. CRITERION RELATED TO AP
We design another criteria on the basis of error reduction for active learning [40] such that once new video set $\mathcal{X}_A$ is selected, it minimize the error of generalization of the new hash functions. Recall that in the hashing model of video with structure information, we learn $K$ hash function by optimizing the average precision over training videos with weak ranking (relevant and non-relevant). Therefore, the winner videos in active set $\mathcal{X}_A$ should increase the performance of average precision. Combining the similarity measurement of video with structure information, in this paper, we adopt the precision-oriented strategy in [39], [41] for active selection of unlabeled videos. This strategy have shown its significant effectiveness in the performance of MAP for image retrieval [39].

We compute the similarity between unlabeled video $X$ and any labeled videos $X_l$ involved in subset $\mathcal{X}_l$ based on cosine-similarity $S(X, X_l)$ and then rank the labeled videos according to these similarities, denoted by $\hat{Y}$. Let the ground truth rank for these labeled videos be $Y^*$. It becomes feasible to compute the average precision $AP(\hat{Y}^*, \hat{Y})$ and the AP loss $A_2(X) = 1 - AP(\hat{Y}^*, \hat{Y})$.

This criterion based on AP measures the informativeness of unlabeled video intuitively. The bigger the value of $AP(\hat{Y}^*, \hat{Y})$, the more opportunities to increase the average precision when video $X$ is labeled.

Based on the criterion formulated in Eq (19), the total AP loss of the selected videos is summarized as $\sum_{x \in \mathcal{X}_A} A_2(X)$. Therefore, we select the active set of videos by solving the following optimization problem

$$\min_{\mathcal{X}_A} \sum_{x \in \mathcal{X}_A} A_2(X) \quad \text{s.t.} \quad |\mathcal{X}_A| = L$$

Combining the criteria related to certainty and AP, our final strategy for active video selection is achieved via considering a trade-off between getting the closest videos and optimizing the average precision, i.e., joint active selection by solving the optimization problem as follows:

$$\min_{\mathcal{X}_A} \sum_{x \in \mathcal{X}_A} \rho A_1(X) + (1 - \rho) A_2(X) \quad \text{s.t.} \quad |\mathcal{X}_A| = L$$

where $\rho > 0$ is a trade-off parameter. This optimization problem can be solved greedily due to the finite number of videos in the pool. Since we select the active videos according to both factors of the certainty and average precision loss, if video $X \in \mathcal{X}_A$ is firstly selected and labeled as positive for example, it is the closest to positive labels and will be well ranked without bring irrelevant videos in the top of ranking. Finally, we well request the users to label all active videos selected in $\mathcal{X}_A$ and update the labeled data set $\mathcal{X}$ to retain the supervised video hashing model with structure information. The alternative process of learning video hash functions with structure information and active selection of unlabeled videos can be repeated for several iterations.

**Algorithm 1 Active Video Hashing With Structure Information**

**Input:** Training data set $\mathcal{Z}$, initial labeled dataset $\mathcal{X}$ and the corresponding ground truth ranking; Trade-off parameters $\lambda, \rho > 0$; Accuracy threshold $\epsilon > 0$.

**Output:** Hashing projection matrix $W$.

1. Supervised video hashing over labeled dataset $\mathcal{X}$:
2. Solve the optimization problem (14) for $W$.
3. Joint active selection:
4. Solve the optimization problem (21) for $\mathcal{X}_A$.
5. Label videos in $\mathcal{X}_A$ and update $\mathcal{X} = \mathcal{X} \cup \mathcal{X}_A$.
6. Repeat step 1 to 5 for several iterations.
C. OPTIMIZATION PROCEDURE

In this section, we focus on solving the proposed challenging optimization problem (16) with cutting-plane algorithm. This method alternatively updates the hash transformation matrix $W$ and the most violated batch of constraints. The details of this algorithm is described in Algorithm 2, where the key issue lies in solving the following optimization problem

$$\min_{W, \xi > 0} \xi + \lambda R(W)$$

s.t. $\forall (Y_1, Y_2, \ldots, Y_N) \in C$

$$\frac{1}{N} \sum_{i=1}^{N} \delta_{\phi}(X_i, Y_i^*) \geq \frac{1}{N} \sum_{i=1}^{N} \nabla(Y_i^*, Y_i) - \xi$$

where $R(W) = ||W||_{2,1} + T(W)$ and finite set $C$ collects the most violated batch of constraints. According to the Pegasos algorithm proposed in [42] for traditional SVM, we solve optimization problem (22) by alternating between stochastic sub-gradient descent and projection steps, where the sub-gradient descent is performed iteratively by picking the most violated ranking at each iteration picks to minimize the slack variable.

Considering the non-smoothness of $\ell_{2,1}$-norm and MAP for image retrieval $\ell_{\infty}$-norm used in $R(W)$, we following the Nesterov’s smooth approximation [43] and approximate the term of $R(W)$ by

$$R_\mu(W) = \sum_{i=1}^{d} f_{i, \mu}(W) + \sum_{j=1}^{T} g_{j, \mu}(W)$$

(22)

where the approximation function $f_{i, \mu}(W)$ and $g_{j, \mu}(W)$ are differentiable and their gradient is Lipschitz continuous with the constant $1/\mu$. They are defined respectively as

$$f_{i, \mu}(W) = \max_{||v||_2 \leq 1} (W^T_i v) - \frac{\mu}{2} ||v||_2^2$$

(23)

$$g_{j, \mu}(W) = \max_{||u||_1 \leq 1} (W^T X_j u) - \frac{\mu}{2} ||u||_2^2$$

(24)

for $i = 1, 2, \ldots, d$ and $j = 1, 2, \ldots, T$, where $\mu > 0$ is a smoothness parameter to control the accuracy of the approximation. $W_i$: denotes the $i$-th row of matrix $W$. Vectors $v, u \in \mathbb{R}^K$ correspond to the auxiliary variables of $W_i: \in \mathbb{R}^{1 \times K}$ and $(W^T X_j)_{j} \in \mathbb{R}^K$, respectively.

As a result, the update for $W$ at each iteration of Pegasos algorithm is given by

$$W \leftarrow (1-\eta \lambda) \frac{\partial R_\mu(W)}{\partial W} + \eta \frac{1}{N} \sum_{i=1}^{N} W(K(X_i, Y_i^*) - K(X_i, Y_i))$$

(25)

Subsequently, we give the derivation of the gradient of regularizer $R_\mu$. When the variable $W$ is fixed, the unique maximizer of optimization problem (23) is derived by

$$\nu(W_{i,:}) = \arg \max_{||v||_2 \leq 1} (W_{i,:}^T v) - \frac{\mu}{2} ||v||_2^2 = \left[ W_{i,:} / \mu \right] \min(\mu, ||W_{i,:}||_2).$$

(25)

Algorithm 2 Video Hashing With Structure Information

Input: Video training set $X$, ranking $Y_1^*, Y_2^*, \ldots, Y_N^*$, trade-off parameters $\lambda > 0$ and accuracy threshold $\epsilon > 0$.

Output: Transformation matrix $W$ and slack variable $\xi \geq 0$

1: Initialize the constraints set $C \leftarrow \emptyset$
2: repeat
3: solve for the optimal $W \in \mathbb{R}^{d \times K}$ and slack $\xi \geq 0$ by
$$(W, \xi) \leftarrow \arg \min_{W, \xi > 0} \xi + R(W) \text{s.t.} (Y_1, Y_2, \ldots, Y_N) \in C$$
4: for $i = 1, 2, \ldots, N$ do
5: $\hat{Y}_i \leftarrow \arg \max_{y \in Y} \nabla(Y^*, Y) + \Psi(X_i, Y)$
6: end for
7: $C \leftarrow C \cup \{(\hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_N)\}$
8: until

$$\frac{1}{N} \sum_{i=1}^{N} \nabla(Y_i^*, \hat{Y}_i) - \frac{1}{N} \sum_{i=1}^{N} \delta_{\phi}(X_i, Y_i^*) \leq \xi + \epsilon$$

As a result, the gradient of function $f_{i, \mu}(W)$ is $\nu(W_{i,:}) \in \mathbb{R}^K$ for $i = 1, 2, \ldots, d$, and the gradient of $\sum_{i=1}^{N} f_{i, \mu}(W)$ is formuluated by

$$\nu(W) = \frac{\partial}{\partial W} \sum_{i=1}^{N} f_{i, \mu}(W)$$

$$= [\nu(W_{1,:}), \nu(W_{2,:}), \ldots, \nu(W_{d,:})] \in \mathbb{R}^{d \times K}$$

(26)

On the other hand, the unique maximizer for optimization problem (24) is calculated using $l_1$-ball projection algorithm in [44], i.e.

$$\nu(W^T \check{X}_{i,:}) = \arg \min_{||u||_1 \leq 1} -((W^T \check{X})_{i,:} - u) + \mu^2/2$$

$$= \arg \min_{||u||_1 \leq \frac{1}{\mu}} (W^T \check{X})_{i,:] - u$$

(27)

for $j = 1, 2, \ldots, T$. The gradient of $g_{j, \mu}(W)$ is derived as

$$\frac{\partial g_{j, \mu}(W)}{\partial W} = \check{X}_{i,j} u \in \mathbb{R}^{d \times K}$$

(28)

where $\check{X}_{i,j} \in \mathbb{R}^d$ is the $j$-th column of matrix $\check{X}$ ($j = 1, 2, \ldots, T$). According to the analysis above, the gradient of $R_\mu(W)$ with respect to $W$ can be calculated by

$$\frac{\partial R_\mu(W)}{\partial W} = \nu(W) + \sum_{j=1}^{T} \check{X}_{i,j} u((W^T \check{X})_{i,:})$$

(29)

V. EXPERIMENTS

In this section, we conduct an extensive set of experiments to demonstrate the superiority of the proposed approach on four widely used benchmark datasets, including the Columbia Consumer Video (CCV) dataset, TRECVID MEDTest 2014, MEDTest 2013 datasets, and UCF 101 dataset.
TABLE 1. Mean average precision (mAP) comparisons of different methods on the CCV and MEDTest 2014 datasets. Performance is reported in percentages.

| Method | CCV 12-bit | CCV 16-bit | CCV 24-bit | CCV 32-bit | CCV 48-bit | CCV 64-bit | MEDTest 2014 12-bit | MEDTest 2014 16-bit | MEDTest 2014 24-bit | MEDTest 2014 32-bit | MEDTest 2014 48-bit | MEDTest 2014 64-bit |
|--------|------------|------------|------------|------------|------------|------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| SH     | 13.8 ± 1.5 | 14.4 ± 1.8 | 14.7 ± 1.7 | 14.9 ± 1.9 | 15.1 ± 1.6 | 15.2 ± 1.8 | 11.6 ± 2.1        | 12.4 ± 1.7        | 12.8 ± 1.9        | 13.4 ± 1.7        | 14.2 ± 1.9        | 15.1 ± 1.7        |
| BRE    | 15.2 ± 1.8 | 17.6 ± 1.6 | 18.3 ± 1.9 | 20.2 ± 1.7 | 21.1 ± 1.8 | 22.5 ± 1.6 | 12.1 ± 2.3        | 12.1 ± 2.1        | 13.5 ± 1.8        | 14.1 ± 1.9        | 14.8 ± 2.2        | 15.4 ± 1.8        |
| SPLH   | 15.4 ± 1.6 | 17.9 ± 1.5 | 18.7 ± 1.7 | 20.8 ± 1.9 | 21.6 ± 1.5 | 23.1 ± 1.7 | 13.4 ± 1.9        | 14.1 ± 2.2        | 14.8 ± 1.8        | 15.3 ± 1.5        | 16.1 ± 1.8        | 15.9 ± 2.1        |
| VHDIT  | 16.8 ± 1.9 | 20.4 ± 1.6 | 21.5 ± 1.8 | 22.4 ± 1.5 | 23.1 ± 1.7 | 23.7 ± 1.8 | 15.1 ± 1.9        | 16.3 ± 2.1        | 16.1 ± 1.9        | 16.9 ± 1.6        | 17.8 ± 1.8        | 18.3 ± 1.9        |
| AH     | 19.4 ± 1.8 | 20.1 ± 1.4 | 21.8 ± 1.7 | 22.4 ± 1.9 | 22.6 ± 1.6 | 22.7 ± 1.4 | 14.6 ± 1.8        | 15.8 ± 2.1        | 15.9 ± 1.7        | 16.3 ± 1.9        | 17.4 ± 2.1        | 17.8 ± 1.8        |
| AHSI   | 20.8 ± 1.4 | 22.1 ± 1.6 | 23.6 ± 1.4 | 24.4 ± 1.2 | 24.9 ± 1.5 | 26.3 ± 1.3 | 16.4 ± 1.5        | 17.9 ± 1.8        | 18.3 ± 1.5        | 18.2 ± 1.4        | 19.4 ± 1.6        | 20.1 ± 1.4        |

TABLE 2. Mean average precision (mAP) comparisons of different methods on the MEDTest 2013 and UCF101 datasets. Performance is reported in percentages.

| Method | MEDTest 2013 12-bit | MEDTest 2013 16-bit | MEDTest 2013 24-bit | MEDTest 2013 32-bit | MEDTest 2013 48-bit | MEDTest 2013 64-bit | UCF101 12-bit | UCF101 16-bit | UCF101 24-bit | UCF101 32-bit | UCF101 48-bit | UCF101 64-bit |
|--------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|----------------|----------------|----------------|----------------|----------------|----------------|
| SH     | 14.9 ± 1.8          | 15.7 ± 2.1          | 16.3 ± 1.5          | 17.2 ± 1.9          | 17.8 ± 2.0          | 18.5 ± 1.8          | 16.8 ± 1.9        | 17.5 ± 1.8        | 18.8 ± 2.1        | 19.6 ± 1.9        | 21.1 ± 2.3        | 22.3 ± 1.7        |
| BRE    | 16.4 ± 1.9          | 18.1 ± 2.1          | 18.9 ± 1.8          | 20.2 ± 2.0          | 21.5 ± 2.2          | 23.2 ± 1.9          | 17.6 ± 1.9        | 18.5 ± 2.4        | 19.4 ± 2.1        | 20.3 ± 2.2        | 22.0 ± 2.4        | 23.1 ± 1.7        |
| SPLH   | 16.8 ± 1.8          | 18.5 ± 1.9          | 19.4 ± 1.8          | 21.1 ± 2.1          | 21.9 ± 1.8          | 22.4 ± 1.6          | 18.1 ± 1.7        | 18.9 ± 2.1        | 20.2 ± 1.7        | 21.4 ± 1.8        | 23.2 ± 2.1        | 23.8 ± 1.9        |
| VHDIT  | 17.4 ± 2.2          | 18.1 ± 1.7          | 18.8 ± 2.0          | 20.3 ± 1.8          | 21.8 ± 2.1          | 22.9 ± 1.9          | 18.8 ± 2.2        | 19.3 ± 1.9        | 21.4 ± 2.1        | 22.5 ± 1.8        | 23.9 ± 1.7        | 24.4 ± 2.0        |
| AH     | 18.5 ± 1.9          | 19.8 ± 1.8          | 20.5 ± 2.0          | 21.3 ± 2.2          | 22.2 ± 1.8          | 22.8 ± 1.9          | 18.4 ± 2.2        | 19.1 ± 2.1        | 20.8 ± 2.1        | 22.1 ± 1.7        | 23.4 ± 2.3        | 24.1 ± 1.9        |
| AHSI   | 20.2 ± 1.5          | 21.8 ± 1.8          | 22.4 ± 1.6          | 23.8 ± 1.4          | 24.4 ± 1.6          | 25.3 ± 1.5          | 21.3 ± 1.6        | 22.7 ± 1.5        | 23.4 ± 1.7        | 24.7 ± 1.9        | 25.3 ± 1.6        | 25.8 ± 1.5        |

A. SETUP

1) DATASETS
We conduct the experiments on four large-scale video datasets, i.e., the Columbia Consumer Video (CCV) [45], TRECVID MEDTest 2014,1 MEDTest 20132 and UCF 101 dataset [46]. For each dataset, we select 10 samples to label for each category.

2) FEATURE EXTRACTION
For each video clip in the datasets, we sample one frame every 2 seconds and extract the frame level CNN descriptors using the architecture of [47]. They key insight in [47] is that by using smaller convolution filters (3 × 3) and very deep architecture (16-19 layers), significant improvement on the ImageNet Challenge 2014 can be achieved. Due to its excellent performance on images, we therefore choose to apply the same architecture to the video datasets.

3) COMPETITORS
To validate the performance of the proposed algorithm, we compare with the following alternatives: (1) Spectral Hashing (SH) [25]: We directly use random sampling + spectral hashing as a baseline. Based on the assumption that data are sampled from a uniform distribution, data are partitioned along their principal directions with the consideration of spatial frequencies; (2) Binary Reconstructive Embedding (BRE) [48]: We first select the most informative samples using the state-of-the-art active learning methods in [9], [49]. Then the hash functions are learnt based on explicitly minimizing the reconstruction error between the metric space and Hamming space; (3) Sequential Projection Learning based Hashing (SPLH) [50]: Similarly, we first select the most informative samples using the state-of-the-art active learning method in [9]. It plays a tradeoff between supervised information and data properties to design robust hash functions, which aims at alleviating the defects from overfitting or insufficient training data; (4) Active Hashing [51]: Active hashing can actively select the most informative labeled pairs for hash function learning. To be more specific, it identifies the most informative points to label and constructs labeled pairs accordingly; (5) Video Hashing with both Discriminative commonality and Temporal consistency (VHDT) [18]: We first select the most informative samples using the state-of-the-art active learning method in [9]. The video hashing formulates a minimization problem over a structure-regularized empirical loss. Specifically, the structure regularization exploits the common local visual patterns and simultaneously preserves the temporal consistency.

4) EVALUATION PROTOCOLS
To make a fair comparison, we use the hamming ranking as the evaluation criteria, which is widely used in the literature. All the video clips in the dataset are ranked according to their Hamming distance from the query video clips and the desired neighbors are returned from the top of the ranked list. To get the quantitative comparison, we adopt Average Precision (AP) as an evaluation metric for Hamming ranking criterion. We calculate the results for each query and report the mean results across all the queries as the final evaluation metric.

1http://nist.gov/itl/iad/mig/med14.cfm
2http://nist.gov/itl/iad/mig/med13.cfm
The proposed method is implemented using Matlab on a PC with an Intel Duo Core i5-2400 CPU 3.1GHz and 8GB RAM. We tuned all the parameters using 5-fold cross validation and selected the best parameters. We tune the trade-off parameter $\lambda$ in the range of $\{0.01, 0.1, 1, 10, 100\}$ and $\rho$ in the range of $\{0.3, 0.4, 0.5, 0.6, 0.7\}$.

### B. PERFORMANCE COMPARISON

In the experiments, we test the performance using hash codes with different length (in the range of 12 to 64 bits). We report performance curves in terms of mean Average Precision (mAP) of different methods in Table 1 and Table 2. From the experimental results, we make the following observations:

- The proposed AHSI method consistently performs much better than the other alternatives, which confirms its effectiveness for large-scale video hashing problems.
- VHTD outperforms the other alternatives by a large margin. This is because VHTD takes advantages of structure information of video data while generating hashing codes.

- Both AH and AHSI performs better than the other baseline methods, which demonstrates the importance of selecting informative samples to label.

### C. SENSITIVITY ANALYSIS

Contrast to other active video hashing algorithms, the proposed method requires only two parameters including $\lambda$ and $\rho$ to be tuned. In the previous experiments, we estimate the best parameters using a five-fold cross validation method. To demonstrate the parameters’ influence on the performance, we conduct experiments to analyze the parameters’ sensitivity in terms of mAP. Experimental results on the four used datasets are reported in Figure 1. The trade-off parameter $\lambda$ is tuned in the range of $\{0.01, 0.1, 1, 10, 100\}$ and $\rho$ is tuned in the range of $\{0.3, 0.4, 0.5, 0.6, 0.7\}$. The range of $\rho$ is decided empirically.

From the experimental results, we can observe that the performance changes differently with respect to different parameters on different datasets. How to decide the optimal values of the parameters is data dependent. This confirms the
importance for us to use cross-validation to choose the best parameter for each data set.

VI. CONCLUSION
In this work, we have introduced a novel video hashing coding method to optimize the mean average precision straightforwardly. The proposed hashing method is built on the framework of active learning. It explicitly concerns the structure information of video clips to learn the optimal hash functions, where the structure information consists of the temporal consistency between successive frames as well as the local visual patterns shared by videos with the same semantic labels. Instead of using the similarity between each pair of videos, ranking-based loss function is utilized in this paper, which directly maximizes the performance of mean average precision. Extensive experiments on three large-scale video datasets have confirmed the superiority of the proposed method.

REFERENCES
[1] X. Chang, Y.-L. Yu, Y. Yang, and E. P. Xing, “Semantic pooling for complex event analysis in untrimmed videos,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 8, pp. 1617–1632, Aug. 2017.
[2] L. Zhang, M. Luo, J. Liu, X. Chang, Y. Yang, and A. G. Hauptmann, “Deep top-k ranking for image–sentence matching,” IEEE Trans. Multimedia, vol. 22, no. 3, pp. 775–785, Mar. 2020.
[3] L. Zhang, X. Yang, J. Liu, M. Luo, S. Wang, Z. Ge, and A. G. Hauptmann, “ZSTAD: Zero-shot temporal activity detection,” in Proc. CVPR. 2020. [Online]. Available: https://arxiv.org/abs/2003.05583
[4] C.-L. Liu, W.-H. Hsiao, C.-H. Lee, T.-H. Chang, and T.-H. Kuo, “Semi-supervised text classification with universal learning,” IEEE Trans. Cybern., vol. 46, no. 2, pp. 462–473, Feb. 2016.
[5] D. Wang, F. Nie, and H. Huang, “Large-scale adaptive semi-supervised learning via unified inductive and transductive model,” in Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD), 2014.
[6] Z. Li, L. Yao, X. Chang, K. Zhan, J. Sun, and H. Zhang, “Zero-shot event detection via event-adaptive concept relevance mining,” Pattern Recognit., vol. 88, pp. 595–603, Apr. 2019.
[7] S. Vijayanarasimhan, P. Jain, and K. Grauman, “Far-sighted active learning on a budget for image and video recognition,” in Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit., Jun. 2010, pp. 3035–3042.
[8] Y. Song, X.-S. Hua, L.-R. Dai, and M. Wang, “Semi-automotive video annotation based on active learning with multiple complementary predictors,” in Proc. 7th ACM SIGMM Int. Workshop Multimedia Inf. Retr. (MIR), 2005, pp. 97–104.
[9] Y. Yang, Z. Ma, F. Nie, X. Chang, and A. G. Hauptmann, “Multi-class active learning by uncertainty sampling with diversity maximization,” Int. J. Comput. Vis., vol. 113, no. 2, pp. 113–127, Jun. 2015.
[10] L. Zhang, J. Liu, M. Luo, X. Chang, Q. Zheng, and A. G. Hauptmann, “Scheduled sampling for one-shot learning via matching network,” Pattern Recognit., vol. 96, Dec. 2019, Art. no. 106962.
[11] M. Luo, F. Nie, X. Chang, Y. Yang, A. G. Hauptmann, and Q. Zheng, “Adaptive unsupervised feature selection with structure regularization,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 4, pp. 944–956, Apr. 2018.
[12] Z. Ma, X. Chang, Z. Xu, N. Sebe, and A. G. Hauptmann, “Joint attributes and event analysis for multimedia event detection,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 7, pp. 2921–2930, Jul. 2018.
[13] Z. Li, F. Nie, X. Chang, L. Nie, H. Zhang, and Y. Yang, “Rank-constrained spectral clustering with flexible embedding,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 12, pp. 6073–6082, Dec. 2018.
[14] Z. Li, F. Nie, X. Chang, Y. Yang, C. Zhang, and N. Sebe, “Dynamic affinity graph construction for spectral clustering using multiple features,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 12, pp. 6323–6332, Dec. 2018.
[15] S. Tong and E. Chang, “Support vector machine active learning for image retrieval,” in Proc. 9th ACM Int. Conf. Multimedia (MULTIMEDIA), 2001, pp. 107–118.
[16] L. Wang, K. Luk Chan, and Z. Zhang, “Booststrapping SVM active learning by incorporating unlabelled images for image retrieval,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Jun. 2003, pp. 192–197.
[17] K. Brinker, “Incorporating diversity in active learning with support vector machines,” in Proc. Int. Conf. Mach. Learn., 2003, pp. 59–66.
[18] G. Ye, D. Liu, J. Wang, and S.-F. Chang, “Large-scale video hashing via structure learning,” in Proc. IEEE Int. Conf. Comput. Vis., Dec. 2013, pp. 2272–2279.
[19] X. Chang, Z. Ma, Y. Yang, Z. Zeng, and A. G. Hauptmann, “Bi-level semantic representation analysis for multimedia event detection,” IEEE Trans. Cybern., vol. 47, no. 5, pp. 1180–1197, May 2017.
[20] X. Chang, Z. Ma, M. Lin, Y. Yang, and A. G. Hauptmann, “Feature interaction augmented sparse learning for fast kinetic motion detection,” IEEE Trans. Image Process., vol. 26, no. 8, pp. 3911–3920, Aug. 2017.
[21] M. Li, “Video hashing: Algorithm design and performance analysis,” Ph.D. dissertation, Dept. Elect. Eng., Pennsylvania State Univ., State College, PA, USA, 2014.
[22] Y. Zhen, Y. Gao, D.-Y. Yeung, H. Xa, and Z. Li, ”Spectral multimodal hashing and its application to multimedia retrieval,” IEEE Trans. Cybern., vol. 46, no. 1, pp. 27–38, Jan. 2016.
[23] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, “Locality-sensitive hashing scheme based on p-stable distributions,” in Proc. SGC, 2004 pp. 131–132.
[24] D. Zhang, J. Wang, D. Cai, and J. Lu, “Self-taught hashing for fast similarity search,” in Proc. 33rd Int. ACM SIGIR Conf. Res. Dev. Inf. Retr. (SIGIR), 2010.
[25] Y. Weiss, A. Torralba, and R. Fergus, “Spectral hashing,” in Proc. Adv. Conf. Neural Inf. Process. Syst., 2009, pp. 1753–1760.
[26] S. Lee and C. D. Yoo, “Robust video fingerprinting for content-based video identification,” IEEE Trans. Circuits Syst. Video Technol., vol. 18, no. 7, pp. 983–988, Jul. 2008.
[27] Q. Wang, L. Si, and B. Shen, “Learning to hash on partial multi-modal data,” in Proc. Int. Conf. Mach. Learn., Jul. 2015, pp. 3904–3910.
[28] M. M. Esmaeili and R. K. Ward, “Robust video hashing based on temporally informative representative images,” in Proc. Dig. Tech. Papers Int. Conf. Consum. Electron. (ICCE), Jan. 2010, pp. 179–180.
[29] A. Joly, O. Buisson, and C. Frelilot, “Content-based copy retrieval using distortion-based probabilistic similarity search,” IEEE Trans. Multimedia, vol. 9, no. 2, pp. 293–306, Feb. 2007.
[30] L. Cao, Z. Li, Y. Mu, and S.-F. Chang, “Submodular video hashing: A unified framework towards video pooling and indexing,” in Proc. 20th ACM Int. Conf. Multimedia (MM), 2012, pp. 299–308.
[31] B. Coskun, B. Sankur, and N. Memon, “Spatio–temporal transform based video hashing,” IEEE Trans. Multimedia, vol. 8, no. 6, pp. 1190–1208, Dec. 2006.
[32] X. Chang and Y. Yang, “Semisupervised feature analysis by mining correlations among multiple tasks,” IEEE Trans. Neural Netw. Learn. Syst., vol. 28, no. 10, pp. 2294–2305, Oct. 2017.
[33] J. Wang, W. Liu, A. X. Sun, and Y.-G. Jiang, “Learning hash codes with listwise supervision,” in Proc. IEEE Int. Conf. Comput. Vis., Dec. 2013, pp. 1792–1800.
[34] S. Chakraborty, V. Balasubramanian, and S. Panchanan, “Adaptive batch mode active learning,” IEEE Trans. Neural Netw. Learn. Syst., vol. 26, no. 8, pp. 1747–1760, Aug. 2015.
[35] S. Mohamad, A. Bouchachia, and M. Sayed-Mouchaweh, “A bi-criteria active learning algorithm for dynamic data streams,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 1, pp. 74–86, Jan. 2018.
[36] H. Liu, Y. Liu, and F. Sun, “Robust exemplar extraction using structured sparse coding,” IEEE Trans. Neural Netw. Learn. Syst., vol. 26, no. 8, pp. 1816–1821, Aug. 2015.
[37] T. Joachims, T. Finley, and C.-N.-J. Yu, “Cutting-plane training of structural SVMs,” Mach. Learn., vol. 77, no. 1, pp. 27–59, Oct. 2009.
[38] F. Wu, X. Lu, Z. Zhang, S. Yan, Y. Rui, and Y. Zhuang, “Cross-media semantic representation via bi-directional learning to rank,” in Proc. ACM MM, 2013, pp. 877–886.
[39] P. H. Gosselin and M. Cord, “Active learning methods for interactive image retrieval,” *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1200–1211, Jul. 2008.

[40] N. Roy and A. McCallum, “Toward optimal active learning through monte carlo estimation of error reduction,” in *Proc. ICML*, 2001, pp. 441–448.

[41] C. Li, Z. Li, Z. Ge, and M. Li, “Knowledge driven temporal activity localization,” *J. Vis. Commun. Image Represent.*, vol. 64, Oct. 2019, Art. no. 102628.

[42] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter, “Pegasos: Primal estimated sub-gradient solver for SVM,” *Math. Program.*, vol. 127, no. 1, pp. 3–30, Mar. 2011.

[43] Y. Nesterov, “Smooth minimization of non-smooth functions,” *Math. Program.*, vol. 103, no. 1, pp. 127–152, May 2005.

[44] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, “Efficient projections onto the $l_1$-ball for learning in high dimensions,” in *Proc. ICML*, 2008, pp. 272–279.

[45] Y.-G. Jiang, G. Ye, S.-F. Chang, D. Ellis, and A. C. Loui, “Consumer video understanding: A benchmark database and an evaluation of human and machine performance,” in *Proc. 1st ACM Int. Conf. Multimedia Retr. (ICMR)*, 2011, p. 29.

[46] K. Soomro, A. Roshan Zamir, and M. Shah, “UCF101: A dataset of 101 human actions classes from videos in the wild,” 2012, *arXiv:1212.0402*. [Online]. Available: http://arxiv.org/abs/1212.0402

[47] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.

[48] B. Kulis and T. Darrell, “Learning to hash with binary reconstructive embeddings,” in *Proc. Conf. Neural Inf. Process. Syst.*, 2009, pp. 1042–1050.

[49] X. Chang, Y. Yang, E. P. Xing, and Y.-L. Yu, “Complex event detection using semantic saliency and nearly-isotonic SVM,” in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, Lille, France, Jul. 2015, pp. 1348–1357.

[50] J. Wang, S. Kumar, and S.-F. Chang, “Semi-supervised hashing for large-scale search,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2393–2406, Dec. 2012.

[51] Y. Zhen and D.-Y. Yeung, “Active hashing and its application to image and text retrieval,” *Data Mining Knowl. Discovery*, vol. 26, no. 2, pp. 255–274, Mar. 2013.

* * *