Video Hashing with Tensor Robust PCA and Histogram of Optical Flow for Copy Detection

Mengzhu Yu¹,², Zhenjun Tang¹,²,³, Hanyun Zhang¹,², Xiaoping Liang¹,² and Xianquan Zhang¹,²

¹Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education, Guangxi Normal University, Guilin 541004, China
²Guangxi Key Lab of Multi-Source Information Mining & Security, Guangxi Normal University, Guilin 541004, China
³Corresponding author: zjtang@gxnu.edu.cn

This paper proposes a novel video hashing with tensor robust Principal Component Analysis (PCA) and Histogram of Optical Flow (HOF) for copy detection. In the proposed hashing, a video is divided into some video groups. For each video group, a low-rank secondary frame is constructed from the low-rank component decomposed by applying tensor robust PCA to the video group. Since the low-rank component can well indicate spatial-temporal intrinsic structure of the video group and it is slightly disturbed by digital operations, feature extraction from the low-rank secondary frames is discriminative and stable. Next, spatial features and temporal features are extracted from low-rank secondary frames by Charlier moments and HOF, respectively. Since the Charlier moments are robust to geometric transform and they can efficiently distinguish video frames with different contents, the use of Charlier moments can make robust and discriminative spatial features. As the HOF can measure the distribution of motion information between frames, the temporal features formed by HOFs can provide good discrimination. Hash is ultimately determined by quantizing the spatial and temporal features and concatenating the quantized results. Numerous experiments on open video datasets indicate that the proposed hashing is superior to some hashing baseline schemes in terms of classification and copy detection.

Keywords: tensor robust principal component analysis (TRPCA); histogram of optical flow (HOF); video hashing; Charlier moments; copy detection

1. INTRODUCTION

In the digital era, the number of digital videos transmitted and stored on the Internet is continuously growing. Meanwhile, with some friendly used editing tools, digital videos are easily processed and re-distributed over social networks. Therefore, it is essential to protect the copyright of digital videos. For example, some social network platforms, such as TikTok, YouTube and Tencent Video, have massive amounts of video data. They urgently need efficient video copy detection schemes to check the copyright of the uploaded videos for avoiding copyright disputes [1–5]. Figure 1 shows an example of a digital video and its several copies, where (a) is a digital video, and (b), (c) and (d) are its copies generated by brightness adjustment, additive white Gaussian noise and adding a Logo, respectively. High-efficiency video copy detection schemes are expected to accurately detect all copies of the query digital video, e.g. all three copies of Fig. 1a. In the past years, some researchers have tried to address copy detection problem by video hashing [6–8]. In this work, an efficient robust video hashing scheme is proposed for copy detection.

Video hashing [9] is a powerful technique for video processing. Based on the visual video content, it can map the input video to a short digit sequence named hash. In practical applications, the use of a hash for representing the video itself can reduce storage cost and speed up similarity computation. Therefore, many useful video hashing schemes have been proposed for various applications [10–12], such as digital watermarking, video classification and video retrieval. To satisfy these applications, video hashing must meet two fundamental properties: robustness and discriminative. Robustness implies that two videos with similar contents must be mapped to the similar or same hashes with a high probability, and discrimination implies that the probability of mapping two different videos to similar hashes must be very low. Most video hashing schemes do not make a satisfactory balance between discrimination and robustness, and thus, their performances in the application of copy detection are not desirable yet.

To address this, a new video hashing with tensor robust Principal Component Analysis (PCA) and Histogram of Optical Flow (HOF) is proposed for video copy detection. Different from the existing video hashing techniques, the proposed video hashing has the below major contributions:

1. A low-rank secondary frame is constructed from the low-rank component decomposed by applying Tensor Robust PCA (TRPCA) to the video group. Since the low-rank component can well indicate spatial-temporal intrinsic structure of video group and it is slightly disturbed by digital operations, feature extraction from low-rank secondary frames is discriminative and stable.

2. Spatial features and temporal features are extracted from low-rank secondary frames by Charlier moments and HOF, respectively. Since the Charlier moments are robust to geometric transformation and they can efficiently distinguish video frames with different contents, the use of Charlier moments can make robust and discriminative spatial features. As the HOF can measure the distribution of motion information between frames, the temporal features formed by HOFs can provide good discrimination.

Extensive experiments on open video datasets are carried out. The results demonstrate that the proposed video hashing scheme...
can maintain a satisfactory balance between discrimination and robustness and outperforms some baseline schemes. In addition, comparative experiments of video copy detection illustrate that our proposed scheme is also superior to the compared schemes. The structure of the rest part of the paper is as follows. Related work on video hashing is described in Section 2. The proposed video hashing scheme is introduced in Section 3. Various experiments are discussed in Section 4. Finally, conclusions are summarized in Section 5.

2. RELATED WORK

Researchers have come up with various video hashing schemes. Depending on the type of their used feature extraction techniques, the reported schemes can be classified into four categories. Typical techniques of each category are described as follows.

2.1. Orthogonal transform-based hashing schemes

These video hashing schemes utilize different orthogonal transforms to extract coefficients of video in spatial and temporal domains for hash construction, including three Dimensional Discrete Wavelet Transform (3D DWT), Temporal Wavelet Transform (TWT), Discrete Sine Transform (DST) and three Dimensional Discrete Cosine Transform (3D DCT). In [13], 3D DWT is exploited to extract the spatial-temporal low-frequency sub-band of video, and the DCT coefficients of the sub-band are calculated to generate hash. This scheme can resist compression and frame dropping. Sandeep et al. [14] present a video hashing scheme with TWT and Random Projections (RP) for near-identical video retrieval. In this scheme, TWT is utilized to calculate representative images, and RP is used to compress representative images for hash generation. The TWT-RP hashing is robust against brightness adjustment and watermark insertion, but it does not resist large-angle rotation. Khelifi et al. [15] design a video hashing scheme with DCT and DST. This hashing scheme can be applied to video recognition and authentication. Tang et al. [8] perform 2D DCT on the grouped frames to construct the feature matrix and adopt non-negative matrix factorization (NMF) to compress the feature matrix for hash generation. The DCT-NMF hashing is resilient to random frame dropping and small-angle rotation.

2.2. Statistical feature-based hashing schemes

These video hashing schemes calculate statistical features in spatial and temporal domains to derive hash, such as mean, variance, histogram and moment. Oostveen et al. [16] propose a video hashing scheme based on mean calculation. They calculate the difference between the block means of frames to construct hash. Since the block means are sensitive to compression and geometric transform, the robustness of their hashing is not ideal. Muceder et al. [17] develop a video hashing scheme with luminance variations. This scheme can effectively identify video attacked by MPEG-4 compression and MPEG-2 compression. Himeur et al. [18] jointly exploit textural and color descriptors to design a video hashing scheme. In this scheme, the Binarized Statistical Image Features (BSIF) of each frame are used to generate BSIF images, and then, the ring partition is used to construct histograms for hash generation. In another study, Himeur et al. [19] adopt the BSIF technique based on ring partition and invariant color descriptors to produce hash. This scheme further improves the robustness against video rotation and flipping, but the discrimination is unsatisfactory. Tang et al. [11] exploit 3D DWT to construct secondary frames from the grouped frames and use the Hu invariant moments to calculate statistical features for hash generation. The DWT-Hu hashing has good robustness.

2.3. Visual feature point-based hashing schemes

These video hashing schemes extract visual feature points of video in spatial and temporal domains to generate hash, such as Harris detector, Speeded Up Robust Feature (SURF) and Scale Invariant Feature Transform (SIFT). Li et al. [20] utilize the Harris detector to extract spatial-temporal salient points for hash generation. This hashing can resist geometric attacks, but it has a high computational cost. Yang et al. [21] use SURF to extract local feature points of each frame and calculate the number of feature points on Hilbert curves to generate hash. This scheme can resist various operations, such as horizontal flipping, but the discrimination needs to be improved. Since SIFT feature points are invariant to scaling, rotation and translation, Vretos et al. [22] present a video hashing scheme with SIFT and latent Dirichlet allocation. Peng et al. [23] employ SIFT and pyramid histogram of directed gradients to produce hash for copy detection. The recognition rate of this hashing is not satisfactory. In another work, Neelima et al. [24] jointly exploit SIFT and Singular Value Decomposition (SVD) to produce hash. This hashing scheme is resilient to geometric operations, but its discrimination is not satisfactory.

2.4. Data dimensionality reduction-based hashing schemes

These video hashing schemes view a video as high-dimensional data and map the high-dimensional data to low-dimensional space by data dimensionality reduction techniques for hash generation. The commonly used techniques include NMF, Tucker Decomposition (TD) and Low-Rank and Sparse Decomposition (LRSD). Li et al. [25] present a video hashing scheme using parallel factor analysis of TD. Their hashing is robust to blurring, frame rate change, compression and random frame dropping. Motivated by the use of NMF and ring partition in image hashing [26, 27], Nie et al. [12] utilize Spherical Torus (ST) to decompose the video cube.
and construct the spatial-temporal image. Then, NMF is adopted to compress the spatial-temporal image for hash generation. The ST-NMF scheme shows robustness against blurring and noise and it can be applied to near-duplicate detection, but their discrimination is unsatisfactory. In another study, Nie et al. [7] design a hashing scheme with multi-feature fusion using TD for video copy detection. Their scheme is robust to combinational operations, but it is sensitive to additive Gaussian white noise. Sandeep et al. [6] randomly select sub-blocks of the video to form the tensor and adopt the factor matrices of TD to generate hash for near-identical video retrieval. Their hashing is resilient to rotation when the angle is small, but it does not resist inter-frame attacks. Chen et al. [28] present a video hashing scheme using LRSD and DWT. They extract the low-rank components of each frame by LRSD and compress the low-rank components via DWT to generate hash. The LRSD-DWT scheme can resist small-angle rotation and MPEG-4 compression.

Apart from the four categories mentioned above, some techniques are also adopted to design video hashing. Considering that the representative frame extraction is an important step of video hashing, Liu et al. [29] jointly use the static and dynamic visual attention models to compute representative frames in temporal domain. This hashing scheme is robust to random frame addition and frame dropping. In [30], they fuse appearance features and attention features to produce hash. Their hashing can effectively resist various operations, such as Gaussian blur, Gaussian noise and median filtering. In [31], hidden Markov tree and SVD are utilized to derive video hash. This hashing scheme is robust to common operations, such as additive noise and filtering.

The above review illustrates that considerable progress on video hashing has been made. However, most schemes fail to make a desirable classification performance and thus their performances in the application of copy detection are not desirable yet. To address this, a new video hashing scheme is proposed by using TRPCA and HOF. The proposed video hashing scheme can make desirable classification performance and it can be effectively applied to copy detection.

3. PROPOSED VIDEO HASHING SCHEME

The proposed video hashing scheme is made up of five components, and its block diagram is displayed in Fig. 2. The first component is the pre-processing operation which consists of converting the input video to a grayscale version, resizing the grayscale video to $U \times U \times U$ by spatial and temporal resampling operations and filtering the resized video by Gaussian low-pass filtering. The second component is to divide the pre-processed video into some video groups and construct low-rank secondary frames by applying TRPCA to these video groups. The third component is to extract spatial features from low-rank secondary frames via the Charlier moments. The fourth component is to calculate temporal features between low-rank secondary frames by HOF. Finally, the fifth component is the hash generation by combining the quantized versions of the spatial features and the temporal features. Details of these components are described in the below subsections.

3.1. Low-rank secondary frame construction via TRPCA

In this work, we construct some low-rank secondary frames from the pre-processed video for generating compact hash. In detail, frames of the pre-processed video are divided into $C$ video groups. Let $U$ be the integral multiple of $C$. Thus, each video group consists of $K = (U/C)$ frames. In each video group, a low-rank secondary frame is calculated via TRPCA. The detailed calculation process is explained as follows.

Tensor Robust PCA (TRPCA) [32] is the higher order extension of the classical sparse model called Robust PCA, and it is an effective technique for tensor data analysis. TRPCA provides a useful method of processing video data and has been successfully used in many important fields, such as video surveillance, face recognition and background modeling [33-35]. In TRPCA, a tensor data $L$ can be decomposed into the tensors $R$ and $S$ by the following model:

$$L = R + S,$$

(1)

where $R$ and $S$ are the low-rank and the sparse components of $L$, respectively. In the video applications, $R$ generally indicates spatial-temporal intrinsic structure of video, and $S$ represents detailed video information. In general, the formula (1) can be solved via the convex optimization problem [32]

$$\min_{R, S} \|R\|_1 + \lambda \|S\|_1,$$

(2)

where $\|R\|_1$ denotes a new tensor nuclear norm of $R$ and $\|S\|_1$ denotes the $\ell_1$ norm of $S$.

In practice, the convex optimization problem can be solved via the ADMM [36]. Specifically, the augmented Lagrangian function of the formula (2) is defined by the following equation:

$$\|\langle R, S, \mathbf{Y}, \rho \rangle = \|R\|_1 + \lambda \|S\|_1 + \langle \mathbf{Y}, (R + S - L) \rangle + \frac{\rho}{2} \|R + S - L\|_F^2,$$

(3)

where $\mathbf{Y}$ denotes the Lagrangian multiplier tensor, $\rho > 0$ is the penalty parameter, $(\cdot, \cdot)$ is the inner product between two tensors and $\|\cdot\|_F$ is the Frobenius norm. Then, the $R$ and $S$ can be updated by minimizing the augmented Lagrangian function $\mathbf{Y}$ alternately as follows:

$$R_{k+1} = \arg\min_R \|R\|_1 + \frac{\rho \varepsilon}{2} \|R + S_k - L + \frac{\mathbf{Y}_k}{\rho}\|_F^2,$$

(4)

$$S_{k+1} = \arg\min_S \min \|S\|_1 + \frac{\rho \varepsilon}{2} \|R_{k+1} + S - L + \frac{\mathbf{Y}_k}{\rho}\|_F^2.$$  

(5)

And the Lagrange multiplier tensor $\mathbf{Y}$ is updated as follows:

$$\mathbf{Y}_{k+1} = \mathbf{Y}_k + \rho \varepsilon (R_{k+1} + S_{k+1} - L).$$  

(6)

in which $k$ is the iteration index. Note that the initial $R_0 = S_0 = \mathbf{Y}_0 = 0$ and $\varepsilon = 10^{-8}$. The convergence conditions are $\|R_{k+1} - R_k\|_\infty \leq \epsilon$, $\|S_{k+1} - S_k\|_\infty \leq \epsilon$ and $\|R_{k+1} + S_{k+1} - L\|_\infty \leq \epsilon$, in which $\|\cdot\|_\infty$ is the $\ell_\infty$ norm. More details of TRPCA can be referred to [32].

In our work, TRPCA is employed to each video group and the low-rank component is used to calculate the low-rank secondary frame. Since the low-rank component can well indicate spatial-temporal intrinsic structure of video group and it is slightly disturbed by digital operations, the constructed low-rank secondary frame is discriminative and stable. Suppose that $R(u, v, w)$ is the element in the $u$th row and the $v$th column of the $w$th frame in the low-rank component of the $i$th group ($1 \leq u \leq U, 1 \leq v \leq U, 1 \leq w \leq W$).
in which \( I(x, y) \) is the reconstructed version of \( I(x, y) \). This means that an image can be reconstructed from a large number of moments by inverse transformation. Notably, high-order Charlier moments can achieve small reconstruction error [39].

Here, the variances of Charlier moments are utilized to generate spatial features. This is based on the below considerations. Video frames attacked by digital operations can be considered as the frames with ‘noise’. As the Charlier moments are robust to noise, the feature extracted via the Charlier moments can achieve good robustness. Moreover, the Charlier moments can efficiently distinguish frames with different contents and thus make our spatial features discriminative. Concretely speaking, the \( i \)th low-rank secondary frame \( D_i \) is divided into non-overlapping blocks sized \( r \times r \). Let \( U \) be the integral multiple of \( r \). Thus, the number of blocks is \( L = (U/r)^2 \). Suppose that \( B_i^{(g)} \) (1 \( \leq g \leq L \)) is the \( g \)th block of the \( i \)th low-rank secondary frame. The Charlier moments \( F_i^{(D)} \) of \( B_i^{(g)} \) can be calculated by the formula (9), where the order of moments is chosen as \( r \). Next, the variance of \( F_i^{(D)} \) is selected as the block feature which can be computed by the following formula:

\[
S_i^{(D)} = \frac{\sum_{u=1}^{U} \sum_{v=1}^{V} f_i^{(D)}(u, v) - \mu_f^{(D)}(u, v)}{r^2 - 1},
\]

where \( f_i^{(D)}(u, v) \) is the element in the \( u \)th row and the \( v \)th column of \( F_i^{(D)} \), and \( \mu_f^{(D)}(u, v) \) is the mean of \( F_i^{(D)} \), which is determined as follows:

\[
\mu_f^{(D)}(u, v) = \frac{\sum_{g=1}^{L} f_i^{(D)}(u, v)}{r^2}. \tag{14}
\]

Therefore, the moment feature sequence \( N_i \) of the \( i \)th low-rank secondary frame \( D_i \) is available by concatenating all block features

\[
N_i = [S_1^{(D)}, S_2^{(D)}, \ldots, S_L^{(D)}]. \tag{15}
\]

Finally, these moment features of all low-rank secondary frames are concatenated to generate the spatial feature sequence \( G \) as follows:

\[
G = [N_1, N_2, \ldots, N_C]. \tag{16}
\]

The spatial feature sequence \( G \) has \( C \times L \) floating-point numbers.

### 3.3. Temporal feature extraction via HOF

The optical flow was introduced by Horn et al. [43] in 1981. It can capture inter-frame motion information of video by the pixel-wise relation in temporal domain and has been successfully applied.
to many video research tasks, such as quality assessment, object detection and object tracking [44–46]. HOF [47] is a feature descriptor that can measure the distribution of motion information in different orientations. In our proposed video hashing scheme, HOF is utilized to extract temporal features between low-rank secondary frames.

Assume that \( f(x, y, t) \) is the gray value of a pixel in the position \((x, y)\) at the time \(t\). At the time \(t + \Delta t\), the pixel moves to a new position \((x + \Delta x, y + \Delta y)\) and its gray value is denoted as \( f(x + \Delta x, y + \Delta y, t + \Delta t) \). Based on the gray-level constancy assumption, the relation between the two pixels is determined as follows [43]:

\[
 f(x, y, t) = f(x + \Delta x, y + \Delta y, t + \Delta t). \tag{17}
\]

The equation (17) can be rewritten by expanding the right side using Taylor expansion as follows:

\[
 f(x, y, t) = f(x, y, t) + \frac{\partial f}{\partial x} \Delta x + \frac{\partial f}{\partial y} \Delta y + \frac{\partial f}{\partial t} \Delta t + \eta,
\]

where \( \eta \) is the second-order infinitesimal term which can be ignored when \( \Delta t \) approaches zero. Consequently, the optical flow constraint equation can be deduced from equation (18) as follows:

\[
 f_x a + f_y b + f_t = 0, \tag{19}
\]

where \( a = \frac{\partial f}{\partial x} \) and \( b = \frac{\partial f}{\partial y} \) represent the optical flow of a pixel along the horizontal and vertical directions, and \( f_x = \frac{\partial f}{\partial x}, f_y = \frac{\partial f}{\partial y}, f_t = \frac{\partial f}{\partial t} \) denote the partial derivatives of the gray value of a pixel along the horizontal, vertical and time directions, respectively.

Since there are two unknowns \( a \) and \( b \) in equation (19), the Lucas–Kanade algorithm [43] is adopted to compute the optical flow vector \([a, b]^T\) of each pixel in the frame. A visual example of the optical flow map between adjacent frames is displayed in Fig. 3, where (a) and (b) are adjacent frames in a video, and (c) is their optical flow map. It is observed that optical flow can capture the change of video content in temporal domain and thus can be used to describe inter-frame motion information.

To generate the HOF features, the magnitude \( m \) and orientation \( \theta \) of the optical flow vector of each pixel are calculated as follows:

\[
 m = \sqrt{a^2 + b^2}, \tag{20}
\]

\[
 \theta = \tan^{-1} \frac{a}{b}, \tag{21}
\]

in which \( \theta \) is constrained between \(-180^\circ < \theta \leq 180^\circ\). Here, we choose 9 bins and each bin corresponds to 40° (360° is uniformly divided into 9 intervals). Then, the bin value can be determined by summing the magnitudes of those optical flow vectors whose orientations fall in the corresponding interval of the bin. Finally, the histogram is normalized to obtain the HOF features \( Q \).

To extract temporal features via HOF, we calculate the HOF features \( Q \), \( (1 \leq j \leq C - 1) \) between adjacent low-rank secondary frames, and then concatenate these HOF features to generate the temporal feature sequence \( E \) as follows:

\[
 E = [Q_1, Q_2, \ldots, Q_{C-1}]. \tag{22}
\]

Therefore, the temporal feature sequence \( E \) has \( 9(C - 1) \) floating-point numbers.

### 3.4. Hash generation

Our hash is available by quantizing the spatial feature sequence \( G \) and the temporal feature sequence \( E \) and concatenating the quantized results. Specifically, the feature sequence \( G \) has \( C \times L \) elements in total, and its elements are quantized to binary values by the following rule:

\[
 h^{(i)}(i) = \begin{cases} 
 1, & \text{If } G(i) \leq \mu(G) \\
 0, & \text{Otherwise,} 
\end{cases} \tag{23}
\]

in which \( G(i) \) is the \( ith \) element of \( G \) (\( 1 \leq i \leq C \times L \)), \( \mu(G) \) is the mean of all elements of \( G \) and \( h^{(i)}(i) \) is the \( ith \) element of the sequence \( h^{(1)} \). Likewise, the feature sequence \( E \) is also converted to binary sequence \( h^{(2)} \) sized \( 9(C - 1) \) by similar quantization as follows:

\[
 h^{(2)}(j) = \begin{cases} 
 1, & \text{If } E(j) \leq \mu(E) \\
 0, & \text{Otherwise,} 
\end{cases} \tag{24}
\]

in which \( E(j) \) is the \( jth \) element of \( E \) (\( 1 \leq j \leq 9(C - 1) \)), and \( \mu(E) \) is the mean of all elements of \( E \). Consequently, the final video hash is obtained by concatenating the two binary sequences as follows:

\[
 H = [h^{(1)}, h^{(2)}]. \tag{25}
\]

Hence, the hash length is \( Z = C \times L + 9(C - 1) \) bits.

### 3.5. Hash similarity evaluation

The similarity of our hash sequences is measured by the Hamming distance. Let \( H_1 \) and \( H_2 \) be two hash sequences with length \( Z \). The Hamming distance can be expressed by the following equation:

\[
 d_{\text{Hamming}}(H_1, H_2) = \sum_{c=1}^{Z} |h_1(c) - h_2(c)|, \tag{26}
\]

where \( h_1(c) \) is the \( cth \) element of \( H_1 \) and \( h_2(c) \) is the \( cth \) element of \( H_2 \). In general, a smaller \( d_{\text{Hamming}} \) illustrates more similar elements in the two hash sequences. For two unrelated videos, their hash elements are different and the corresponding \( d_{\text{Hamming}} \) should be a large value. In practice, a threshold is generally used to determine similarity.

### 4. EXPERIMENTAL RESULTS

Experimental parameter settings of our proposed scheme are listed below. The selected video size in the pre-processing is \( 256 \times 256 \times 256 \), the number of video groups is 8 and the chosen block size is 64, i.e. \( U = 256, C = 8 \) and \( r = 64 \). Thus, \( L = (U/r)^2 = 16 \), and our hash length is \( Z = C \times L + 9(C - 1) = 191 \) bits. The MATLAB R2019b is employed to implement our proposed scheme. The used desktop PC for coding has the following devices: the CPU is an Intel Core i7-9850H CPU with 2.60 GHz and the capacity of the RAM is 32.0 GB.
**Table 1.** Parameter settings of the selected operations.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Parameter value</th>
<th>Num</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPEG-2</td>
<td>bit rate [100, 1000], step: 100</td>
<td>10</td>
</tr>
<tr>
<td>MPEG-4</td>
<td>compression quality [10, 100], step: 10</td>
<td>10</td>
</tr>
<tr>
<td>FS</td>
<td>ratio ∈ {0.5, 0.75, 0.9, 1.1, 1.5, 2}</td>
<td>6</td>
</tr>
<tr>
<td>FR</td>
<td>angle ∈ {±0.5°, ±1°, ±1.5°, ±2°}</td>
<td>8</td>
</tr>
<tr>
<td>CA</td>
<td>magnitude ∈ {±5, ±10, ±20, ±30}</td>
<td>8</td>
</tr>
<tr>
<td>BA</td>
<td>magnitude ∈ {±5, ±10, ±20, ±30}</td>
<td>8</td>
</tr>
<tr>
<td>RFA</td>
<td>frame number ∈ {20, 15, 10, 5, 2, 1}</td>
<td>6</td>
</tr>
<tr>
<td>RDF</td>
<td>frame number ∈ {20, 15, 10, 5, 2, 1}</td>
<td>6</td>
</tr>
<tr>
<td>GLF</td>
<td>standard deviation ∈ {0.1, 0.15, 0.2, 0.25, 0.3}</td>
<td>10</td>
</tr>
<tr>
<td>AWGN</td>
<td>signal noise ratio ∈ {6, 5, 4, 3, 2, 1}</td>
<td>6</td>
</tr>
<tr>
<td>SPN</td>
<td>density ∈ {0.001, 0.002, 0.005, 0.007, 0.01, 0.02, 0.05, 0.07, 0.1, 0.2}</td>
<td>10</td>
</tr>
</tbody>
</table>

Total 88

4.1. Classification performance

To analyze the classification of our proposed scheme, the open dataset called ReefVid [48] is chosen to conduct experiments, where the videos are in AVI format and have a resolution of 384 × 288. The used topics include: seagrass, ascidians, anemones, algae, black coral, bryozoans, crustaceans, plankton, jellyfish, cleaning station, bioerosion and caves, and the typical videos in the ReefVid are shown in Fig. 4. We use the videos numbered 1–150 to produce similar videos for evaluating hash robustness. Specifically, we apply 11 types of content-preserving operations to the 150 videos, including MPEG-2 compression, MPEG-4 compression, Frame Scaling (FS), Frame Rotation (FR), Contrast Adjustment (CA), Brightness Adjustment (BA), Random Frame Addition (RFA), Random Frame Dropping (RFD), 3 × 3 Gaussian Low-pass Filtering (GLF), Additive White Gaussian Noise (AWGN) and Salt and Pepper Noise (SPN). The parameter settings of these operations are presented in Table 1. In summary, there are 88 operations that produce 150 × 88 = 13 200 pairs of similar videos. Therefore, the dataset for robustness analysis has 13 200 + 150 = 13 350 videos. The videos numbered 151–350 in the ReefVid are employed to analyze the discrimination. Thus, 200 hash sequences are produced from these videos and then C_{100}^{2} = 200 × (200 − 1)/2 = 19 900 Hamming distances can be obtained.

To validate the classification performance of our proposed scheme, the famous tool called ROC graph [49] is used. The ROC curve is made up of some discrete points whose coordinates are formed by the false positive rate V_{f} and the true positive rate V_{t}. Note that V_{t} indicates discrimination, and V_{f} indicates robustness. The formulas of V_{t} and V_{f} are given by the followings:

\[
V_t = \frac{n_{\text{right}}}{N_{\text{similar}}} \quad (26)
\]

\[
V_f = \frac{n_{\text{wrong}}}{N_{\text{different}}} \quad (27)
\]

where n_{right} represents the number of similar videos successfully identified, n_{wrong} indicates the number of different videos incorrectly distinguished and N_{similar} and N_{different} represent the numbers of similar and different videos, respectively. Therefore, the nearer the curve to the top-left corner, the better the classification performance. That is to say, the larger the area under the curve (AUC), the higher the classification accuracy. Note that AUC value falls in the range [0, 1].

To illustrate classification advantage, our proposed scheme is compared with several baseline video hashing schemes, including ST-NMF scheme [12], TD scheme [6], TWT-RP scheme [14], LRSD-DWT scheme [28], DCT-NMF scheme [8] and DWT-Hu scheme [11]. All videos are converted to 256 × 256, and then input to the compared schemes. Notably, the similarity metrics and experimental parameter settings of all schemes are in line with their source papers. For similarity metric, TD scheme, LRSD-DWT scheme, DCT-NMF scheme and our proposed schemes adopt the Hamming distance, ST-NMF scheme and DWT-Hu scheme adopt the Euclidean distance, and TWT-RP scheme adopts the normalized Hamming distance.

Figure 5 depicts the ROC curves of these schemes. As can be seen, all curves in the ROC graph are around the top-left corner. Obviously, the curve nearest to the top-left corner is given by our proposed scheme. This can be deduced that our proposed scheme has the best classification. In order to further illustrate this, the AUCs of different video hashing schemes are listed in Table 2. The AUCs of our proposed scheme, ST-NMF scheme, TD scheme, TWT-RP scheme, LRSD-DWT scheme, DCT-NMF scheme and DWT-Hu scheme are 0.99998, 0.91923, 0.99960, 0.99910, 0.99241, 0.99081 and 0.99532, respectively. Clearly, the AUC result of our proposed scheme is greater than the AUC results of other baseline schemes. This proves that our proposed scheme is better than other schemes in classification. Our proposed scheme can achieve better classification performance. This can be understood as follows. Our proposed scheme constructs low-rank secondary frames via TRPCA. Since the low-rank component can well indicate spatial-temporal intrinsic structure of video group and it is slightly disturbed by digital operations, feature extraction from low-rank secondary frames is discriminative and stable. In addition, our proposed scheme exploits Charlier moments and HOF to extract spatial features and temporal features, respectively. The use of Charlier moments can make robust and discriminative spatial features, while the temporal features formed by HOFs can provide good discrimination.
Table 2. Performance comparisons.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>AUC</th>
<th>Time (s)</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST-NMF scheme [12]</td>
<td>0.91923</td>
<td>4.41</td>
<td>2048</td>
</tr>
<tr>
<td>TD scheme [6]</td>
<td>0.99960</td>
<td>2.35</td>
<td>512</td>
</tr>
<tr>
<td>TWT-RP scheme [14]</td>
<td>0.99910</td>
<td>2.84</td>
<td>128</td>
</tr>
<tr>
<td>LRSD-DWT scheme [28]</td>
<td>0.99241</td>
<td>9.44</td>
<td>128</td>
</tr>
<tr>
<td>DCT-NMF scheme [8]</td>
<td>0.99081</td>
<td>0.97</td>
<td>128</td>
</tr>
<tr>
<td>DWT-Hu scheme [11]</td>
<td>0.99532</td>
<td>2.33</td>
<td>720</td>
</tr>
<tr>
<td>Our proposed scheme</td>
<td>0.99998</td>
<td>3.38</td>
<td>191</td>
</tr>
</tbody>
</table>

Figure 5. Comparison of different schemes using ROC graph.

Computation time and hash length are also two performance metrics for evaluating schemes. Their results are given in Table 2, where the time is the mean time of computing a hash sequence. The time of our proposed scheme, ST-NMF scheme, TD scheme, TWT-RP scheme, LRSD-DWT scheme, DCT-NMF scheme and DWT-Hu scheme is 3.38, 4.41, 2.35, 2.84, 9.44, 0.97 and 2.33 seconds, respectively. Our proposed scheme has a faster computation time than ST-NMF and LRSD-DWT schemes. The hash lengths of our proposed scheme, ST-NMF scheme, TD scheme, TWT-RP scheme, LRSD-DWT scheme, DCT-NMF scheme and DWT-Hu scheme are 191, 2048, 512, 128, 128, 720 bits, respectively. Apparently, the length of our proposed scheme is less than the lengths of DWT-Hu, TD and ST-NMF schemes, but it is longer than the lengths of other schemes.

4.2. Video copy detection performance

To evaluate our performance in the application of copy detection, we conduct various specific experiments using the open dataset HMDB51 [50]. This dataset consists of 51 video categories, and each category contains at least 50 videos. Their frame resolutions are all 320 × 240. In the experiments, we select 10 video categories from this dataset and then take 30 videos from each category. The themes of the used categories include: catch, climb, dribble, golf, hug, ride bike, push, ride horse, pour and shoot gun. To construct the dataset for copy detection, one video randomly chosen from each category is used as the query video. Next, 10 types of normal digital operations are applied to each query video for producing video copies. The selected parameter values of the 10 operations include the followings:

- MPEG-4 with the compression quality of 100
- FS with the ratio of 2
- FR with the rotation angle of 2°
- CA with the magnitude of 50
- BA with the magnitude of 50
- RFA with the frame number of 20
- RFD with the frame number of 20
- AWGN with the signal noise ratio of 6
- Picture-in-picture with a size of 40% of the video frame size

As there are 10 operations in total, 10 video copies are generated from each query video. Therefore, a dataset of 400 videos is constructed by adding the 100 copies mentioned above to the dataset with 300 original videos.

The Precision-Recall (P-R) graph is selected as the evaluation criterion. In the graph, the vertical axis represents the precision and the horizontal axis indicates the recall. The P-R curve is composed of a series of points whose coordinates are formed by the recall and the precision. The equations of the precision $P_r$ and the recall $W_r$ are given by the following equations:

$$W_r = \frac{P_{\text{correct}}}{P_{\text{returned}}}$$
$$P_r = \frac{P_{\text{correct}}}{P_{\text{copies}}}$$

where $P_{\text{correct}}$ indicates the number of successfully recognized video copies, $P_{\text{returned}}$ represents the number of all results returned and $P_{\text{copies}}$ is the number of all video copies. Therefore, the closer the curve to the top-right corner, the better the copy detection performance. That is to say, the larger the P-R AUC (PRAUC), the higher the copy detection accuracy.

Here, two experiments are performed to verify effectiveness of our proposed scheme in copy detection. These experiments include: (i) comparison of P-R curves and the PRAUCs among different schemes and (ii) the precision/recall comparison under the optimal threshold among different schemes. Detailed experiments are described below.

Our advantage in copy detection is firstly demonstrated by comparing the P-R curves of some baseline schemes, including ST-NMF scheme [12], TD scheme [6], TWT-RP scheme [14], LRSD-DWT scheme [28], DCT-NMF scheme [8] and DWT-Hu scheme [11]. The curves of the compared schemes are displayed in Fig. 6. As can be seen, our curve is nearer to the top-right corner than the curves of the baseline schemes. This visual result illustrates that our proposed scheme holds the best copy detection performance. The PRAUC values of our proposed scheme, ST-NMF scheme, TD scheme, TWT-RP scheme, LRSD-DWT scheme, DCT-NMF scheme and DWT-Hu scheme are 0.9951, 0.6945, 0.9716, 0.9694, 0.8742, 0.9899 and 0.7465, respectively. It is obvious that our proposed scheme has the largest PRAUC, which further validates that our proposed scheme is superior to some baseline schemes in the application of copy detection.

Figure 7 presents the curves of the precision and recall of our video hashing scheme with different thresholds, where the vertical axis indicates the precision/recall and the horizontal axis represents the threshold. It can be observed that as the threshold changes from 0 to 160, the recall increases and the precision decreases. Note that the intersection of the recall and precision curves corresponds to the optimal threshold. Under this threshold, the obtained recall and precision are the same. For our proposed scheme, the optimal threshold is 31 and the corresponding precision/recall is 0.9700. This means that 97.00%
Selection of group number implies better robustness. Likewise, when $C$ values are given in Table 4, the length of $V_f$ is 791 bits and keep other parameters constant. The $V_f$ values are 0.9837, 0.9862, 0.9801 and 0.9754, respectively. The $V_f$ of $C = 8$ is larger than those of other $C$ values. When $V_f \approx 1$, the $V_f$ values of $C = 4$, $C = 8$, $C = 16$ and $C = 32$ are 0.0305, 0.0269, 0.0264 and 0.1459, respectively. The $V_f$ of $C = 8$ is a little larger than that of $C = 16$. Taken together, we choose $C = 8$ to achieve a good classification performance and a short hash length.

4.4. Choice of block size

The choice of block size $r \times r$ is discussed. Specifically, we only change $r$ and keep other parameters constant. Figure 6 displays the ROC curves with various block sizes, where $r$ is chosen from the set of $\{128, 64, 32\}$. We do not choose smaller $r$ values because it will result in too long hash length. Clearly, the best curve is given by $r = 64$. The curve of $r = 128$ is under the curve of $r = 64$, and the curve of $r = 32$ is under the curve of $r = 128$. When $r = 32$, its AUC is 0.99970. When $r = 64$, its AUC increases to 0.99988. When $r = 128$, the AUC does not increase but slightly decreases to 0.99988. The AUC of $r = 64$ is the largest one among these AUC values. This demonstrates that the best classification performance can be achieved when $r = 64$. Performances of various block sizes are reported in Table 4. The hash length of $r = 32$ is 575 bits, the length of $r = 64$ is 191 bits and the length of $r = 128$ is 95 bits. As the $r$ value increases, the hash length decreases. The time of $r = 32$ is 3.35 seconds, the time of $r = 64$ is 3.38 seconds and the time of $r = 128$ is 3.41 seconds. The time of various block sizes changes slightly. Taken together, the best whole performance of our video hashing scheme can be achieved when $r = 64$.

4.5. Selection of orthogonal moments

Our proposed video hashing scheme utilizes the Charlier moments to extract spatial features. To verify the superiority of our choice of the Charlier moments, some existing discrete orthogonal moments are compared with the Charlier moments, including the Meixner moments [39], the Tchebichef moments [38], the Charlier moments [39] and the Krawtchouk moments [40]. We only change the used moments and keep other parameters constant, and hence, the hash lengths of different moments are all 191 bits. The ROC curves with these discrete orthogonal moments are presented in Fig 9, where the top-left corner is enlarged for better viewing. We observe that the top-three curves are provided by the Charlier, Meixner and Krawtchouk moments, respectively. This implies that they all reach good classification results. In addition, the best curve is given by the Charlier moments. The AUCs of the Meixner, Tchebichef, Charlier and Krawtchouk moments are 0.99995,

<table>
<thead>
<tr>
<th>$C$</th>
<th>$V_f$ when $V_f \approx 0$</th>
<th>$V_f$ when $V_f \approx 1$</th>
<th>Length (bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.9837</td>
<td>0.0305</td>
<td>91</td>
</tr>
<tr>
<td>8</td>
<td>0.9862</td>
<td>0.0269</td>
<td>191</td>
</tr>
<tr>
<td>16</td>
<td>0.9801</td>
<td>0.0264</td>
<td>391</td>
</tr>
<tr>
<td>32</td>
<td>0.9754</td>
<td>0.1459</td>
<td>791</td>
</tr>
</tbody>
</table>

Table 4. Performances of various $r$ values.

<table>
<thead>
<tr>
<th>$r$</th>
<th>AUC</th>
<th>Length (bit)</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>0.99988</td>
<td>95</td>
<td>3.41</td>
</tr>
<tr>
<td>64</td>
<td>0.99998</td>
<td>191</td>
<td>3.38</td>
</tr>
<tr>
<td>32</td>
<td>0.99970</td>
<td>575</td>
<td>3.35</td>
</tr>
</tbody>
</table>

Figure 6. Comparison of copy detection using P-R curves.

Figure 7. Precision/recall values under different thresholds.

video copies are correctly identified at the optimal threshold. Likewise, we calculate the optimal threshold and the corresponding precision/recall of each compared scheme. The precision/recall values of ST-NMF scheme, TD scheme, TWT-RP scheme, LRSD-DWT scheme, DCT-NMF scheme and DWT-Hu scheme are 0.6700, 0.9192, 0.9293, 0.8081, 0.9689 and 0.7071, respectively. It is obvious that the precision/recall of our proposed scheme is bigger than those of other hashing schemes under the optimal threshold. This also verifies that the copy detection performance of our proposed scheme outperforms those of other schemes.

4.3. Selection of group number

During low-rank secondary frame construction, the group number is utilized to determine the number of low-rank secondary frames. Hence, the selection of the group number $C$ is discussed. We only change $C$ and keep other parameters constant. The performance comparisons under different $C$ values are given in Table 3. In this table, $V_f$ when $V_f \approx 0$ represents the maximum correct detection rate of similar videos under the condition that almost no different videos are incorrectly classified. Hence, a larger $V_f$ implies better robustness. Likewise, $V_f$ when $V_f \approx 1$ represents the minimum false detection rate of different videos under the condition that similar videos are almost correctly recognized. Therefore, a smaller $V_f$ means better discrimination. It can be found that, when $V_f \approx 0$, the $V_f$ values of $C = 4$, $C = 8$, $C = 16$ and $C = 32$ are 0.9837, 0.9862, 0.9801 and 0.9754, respectively. The $V_f$ of $C = 8$ is larger than those of other $C$ values. When $V_f \approx 1$, the $V_f$ values of $C = 4$, $C = 8$, $C = 16$ and $C = 32$ are 0.0305, 0.0269, 0.0264 and 0.1459, respectively. The $V_f$ of $C = 8$ is a little larger than that of $C = 16$. Taken together, we choose $C = 8$ to achieve a good classification performance and a short hash length.
0.99945, 0.99998 and 0.99996, respectively. The AUC of the Charlier moments is larger than those of other moments. Therefore, we choose the Charlier moments to extract spatial features for achieving better classification performance.

5. CONCLUSIONS
A new video hashing scheme using TRPCA and HOF has been proposed for copy detection. In the proposed hashing, the preprocessed video is divided into some video groups. For each video group, a low-rank secondary frame is constructed from the low-rank component decomposed by applying TRPCA to the video group. Since the low-rank component can well indicate spatial-temporal intrinsic structure of video group and it is slightly disturbed by digital operations, feature extraction from low-rank secondary frames is discriminative and stable. Spatial features and temporal features are extracted from low-rank secondary frames by Charlier moments and HOF, respectively. Since the Charlier moments are robust to geometric transformation and they can efficiently distinguish video frames with different contents, the use of Charlier moments can make robust and discriminative spatial features. As the HOF can measure the distribution of motion information between frames, the temporal features formed by HOFs can provide good discrimination. Numerous experiments have been conducted on open video datasets to prove the efficiency of the proposed video hashing scheme. Performance comparisons have shown that our proposed scheme is superior to some hashing baseline schemes in terms of classification and copy detection.

ACKNOWLEDGEMENTS
Many thanks to the referees for their helpful suggestions.

FUNDING
Guangxi Natural Science Foundation (2022GXNSFAA035506); National Natural Science Foundation of China (62272111, 61962008, 62062013, 62162006); Guangxi ‘Bagui Scholar’ Team for Innovation and Research; Guangxi Talent Highland Project of Big Data Intelligence and Application; Guangxi Collaborative Innovation Center of Multi-source Information Integration and Intelligent Processing.

DATA AVAILABILITY
The datasets used to support the findings of this paper can be downloaded from the public websites which can be found in the cited references.

REFERENCES