

## 基于感知哈希和切块的视频相似度检测方法

吴悦<sup>1,2</sup>, 雒江涛<sup>2\*</sup>, 刘锐<sup>1,2</sup>, 胡钟尹<sup>1</sup>

(1. 重庆邮电大学 通信与信息工程学院, 重庆 400065; 2. 重庆邮电大学 电子信息与网络工程研究院, 重庆 400065)

(\* 通信作者电子邮箱 luojt@cqupt.edu.cn)

**摘要:** 长期以来视频侵权问题层出不穷, 而检测视频相似度是视频侵权的重要手段。针对现有视频相似度检测方法中存在的多特征关系难以关联、时间复杂度高等问题, 提出一种基于感知哈希和切块的快速对比方法。首先, 利用视频的关键图像帧生成数字指纹集; 然后, 基于切块的方法构建相应的倒排索引, 提高数字指纹间的对比速度; 最后, 根据得到的数字指纹间的汉明距离进行相似度判定。实验结果表明, 与传统的感知哈希对比方法相比, 该方法在保证检测准确度的前提下将检测时间平均缩短 93%; 与多特征哈希 (MTH)、自学习哈希 (STH)、光哈希 (SPH) 等三种常见方法相比, 所提方法的平均准确率均值 (mAP) 分别提高了 1.4%、2% 和 2.3%, 检测时间分别缩短了 24%、32% 和 16%, 验证了所提方法的可行性。

**关键词:** 视频相似度; 感知哈希; 切块; 数字指纹; 倒排索引

**中图分类号:** TP391.4 **文献标志码:** A

### Video similarity detection method based on perceptual hashing and dicing

WU Yue<sup>1,2</sup>, LUO Jiangtao<sup>2\*</sup>, LIU Rui<sup>1,2</sup>, HU Zhongyin<sup>1</sup>

(1. School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China;

2. Electronic Information and Networking Research Institute, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

**Abstract:** For a long time, video copyright infringement problems have emerged one after another, and the detection of video similarity is an important approach of identifying video copyright infringement. Concerning the problems of the correlation difficulty of multi-feature relation and high time complexity in the existing video similarity detection methods, a fast comparison method based on perceptual hashing and dicing was proposed. First, the key image frames of the video were used to generate a digital fingerprint set. Then, based on the dicing method, the corresponding inverted index was generated to speed up the comparison between digital fingerprints. Finally, the similarity was judged according to the obtained Hamming distance between the digital fingerprints. Experimental results show that the proposed method can reduce the detection time by an average of 93% with ensuring the detection accuracy compared to the traditional perceptual hashing comparison methods; in the comparison with three common methods including Multi-Feature Hashing (MFH), Self-Taught Hashing (STH) and SPectral Hashing (SPH), the mean Average Precision (mAP) of the proposed method is increased by 1.4%, 2% and 2.3%, respectively, and the detection time is shortened by 25%, 32% and 16%, respectively, which verifies the feasibility of the proposed method.

**Key words:** video similarity; perceptual hashing; dicing; digital fingerprint; inverted index

## 0 引言

视频应用在近年来呈现爆发式增长, 但是类似于涉案金额高达 2.3 亿元的“2·15”系列电影侵权案的视频侵权行为却层出不穷。由于鉴别视频侵权问题的关键是检测两个视频的相似度, 因此只要能够准确检测两段视频的相似度, 则能有效支持侵权的鉴别。除了鉴别视频侵权问题外, 通过视频相似度检测方法可以得到多张图像帧在时空与信息上的相关性与互补性, 以此生成质量较好的融合图片, 因此解决视频相似度检测问题也为图像融合的理论研究奠定了良好的基础。

现有的相似视频检测方法主要分为视频层次检测、图像层次检测和音频层次检测三类方法。

在视频层次检测方法中, 检测两个视频是否完全相同的最

基本的方法就是利用 MD5 信息摘要算法<sup>[1]</sup>, 对比两个视频的 MD5 值是否相等, 但是该方法无法检测到篡改后的视频。基于标题、标签和描述内容的检测方式<sup>[2-4]</sup>则通过人为对视频内容进行高度概括, 后续利用机器学习的方法对视频进行分类; 但该方法需要大量的人工辅助才能完成, 并且由于每个人对视频的理解不同, 产生的标签和描述也有所不同, 导致最终的鉴别效果做不到客观一致。因此大多数基于视频层面的相似视频研究是通过学习结合外观和时间等特征<sup>[5-7]</sup>来进行的。例如, 文献[5]中结合时间和如颜色、纹理等感知视觉特征以及一个计算索引和查询文档之间逻辑推断的匹配框架; 但由于考虑了时间、视觉特征、时间和特征之间的对应关系三方面的因素, 所以计算速度比较缓慢, 且上述利用深度学习的方法, 前期需要

收稿日期: 2020-08-06; 修回日期: 2020-10-20; 录用日期: 2020-10-30。

**作者简介:** 吴悦 (1997—), 女, 重庆人, 硕士研究生, 主要研究方向: 机器视觉; 雒江涛 (1971—), 男, 河南郑州人, 教授, 博士生导师, 博士, 主要研究方向: 移动大数据、新一代网络技术、通信网络测试与优化; 刘锐 (1995—), 男, 重庆人, 硕士研究生, 主要研究方向: 机器视觉; 胡钟尹 (1999—), 女, 重庆人, 主要研究方向: 机器视觉。

大量时间和样本训练网络模型,网络复杂、参数繁多。

在图像层次检测方法中,文献[8]中提出了对图像帧进行基于光流的视频复制移动伪造检测算法;但计算复杂度太大,效率太低。基于视频的图像特征来进行相似视频检测的方法,主要是通过对比视频图像帧的形状、颜色以及纹理<sup>[9]</sup>等视觉特征;但这一类方法只针对形状、颜色等图像特征相似的视频,如果相似视频更改了原始视频的颜色、形状,则会使得图像帧特征差异大,影响结果的判断。文献[10]中使用密集连接卷积网络来提取关键帧的深度特征,然后使用分类相关分析的特征融合算法来分析提取的参数特征,从而提高检测精度;但是由于该方法没有采用相应的视频数据集进行训练,导致存在一定漏检情况。目前,大部分的基于图像的视频相似度对比方法都是对图像特征进行哈希编码得到图像或者视频指纹,后续对这些指纹进行相似度度量。在多特征哈希(Multiple Feature Hashing, MFH)<sup>[11]</sup>方法中,作者利用得到的多个特征的哈希码进行加权求和,得到最后的多特征哈希码。文献[12]中通过自学习哈希(Self-Taught Hashing, STH)算法得到代表图片帧的哈希序列,后续使用STH串的欧几里德空间上的余弦相似度构建局部相似度图,得到两视频相似度;文献[13]中提出了一种基于拉普拉斯算子的特征向量的光哈希(Spectral Hashing, SPH)方法,得到哈希序列之后,利用汉明距离计算两视频之间的相似度。但是多特征哈希、自学习哈希以及光哈希三种方法仅仅使用了各单个特征之间的关系,并没有充分利用多特征之间的关系,因此不同特征表示的联系没有挖掘出来,使得最终的检索效果没有达到最优。

在音频层次检测方法中,主要是通过对比两个视频中的图像联合音频或字幕来获得两视频的相似度。基于频谱可见度图来检测音频相似度<sup>[14]</sup>的方法通过可见度图的方式,既可捕获信号的谐波含量,同时又对宽带噪声具有弹性,因此可以测量两音频数据中谐波信号的相似性。文献[15]中提到通过深度残差网络对端到端视频进行字幕识别,但是基于字幕的相似视频检测方法只能适用于有字幕的相似视频对比。

针对上述方法中的多特征间关系难以建立合适的关联、时间复杂度高等局限性,本文基于对比视频最直接的图像特征,设计出一种基于感知哈希(perceptual Hashing, pHash)<sup>[16]</sup>和切块的视频相似度快速检测方法,并分别在自己构造的数据集和公开数据集CC\_WEB\_VIDEO<sup>[17]</sup>进行了对比实验,验证

了该方法有可行性。

## 1 基于切块和感知哈希的方法原理

动态视频由多个图像帧组成,图像作为视频很重要的一部分,是视频最直观的特征。因此,通过视频中关键图像帧的相似度检测可以判断原始视频和被测视频是否为相似视频。

感知哈希(pHash)算法是使用离散余弦变换(Discrete Cosine Transform, DCT)来得到图片的低频成分从而计算出能代表图片的数字指纹(以下简称指纹),及一个64位的pHash二进制序列。作为检测相似图片最常用的算法,如果将其应用在检测相似视频上,在 $n$ 个指纹和 $m$ 个指纹比较时,最多可能会遍历对比 $n \times m$ 次,无法高效地检测相似视频。因此为了提高现有基于感知哈希视频相似度检测方法的速率,本文在传统的基于感知哈希的视频相似度检测方法<sup>[18]</sup>中引入抽屉原理<sup>[19]</sup>的思想,通过切块进而提出一种视频相似度的快速检测方法。该方法的基本思想是通过切块排除完全不可能相似的指纹,减少不必要的对比,从而加快哈希指纹的对比过程。

根据抽屉原理的推广形式,设把 $n \times m + 1$ 个元素划分至 $n$ 个集合( $P_1, P_2, \dots, P_n$ )中,其中 $P_1, P_2, \dots, P_n$ 表示这 $n$ 个集合对应包含的元素个数,则:至少存在某个集合 $P_i$ ,其包含元素个数 $P_i$ 大于或等于 $m + 1$ 。

基于切块的构思,定义两个汉明距离在设定阈值 $H$ 内的指纹为相似指纹,将一个视频指纹平均切成 $d$ 个等长的块。根据上述推广,两个相似的指纹至少有 $d - H$ 块是完全相等的,可以得出:有 $d - H$ 块完全相等的两指纹有可能相似,但 $d - H$ 块不完全相等的一定不可能相似,从而排除一些不可能相似的指纹。

对比过程如图1所示。其中,图1中倒排索引表中的Value代表某一个组合方式中指纹块的值,Id则代表包含该值的指纹编号。对切分的 $d$ 个指纹块随机取出 $d - H$ 块,对这些指纹块组合形成的值构建倒排索引(从值Value即能找到包含该值的所有的完整指纹Id)。把被测视频的指纹同样切分、随机取出 $d - H$ 块,去倒排索引里查找这些指纹块组合形成的值是否存在:存在即说明该完整指纹可能与包含其的完整指纹相似,进而把两完整指纹进行对比,汉明距离小于 $H$ 的即为相似指纹(关键帧);不存在则说明这两组完整指纹一定不相似,再比较下一指纹块形成的值。

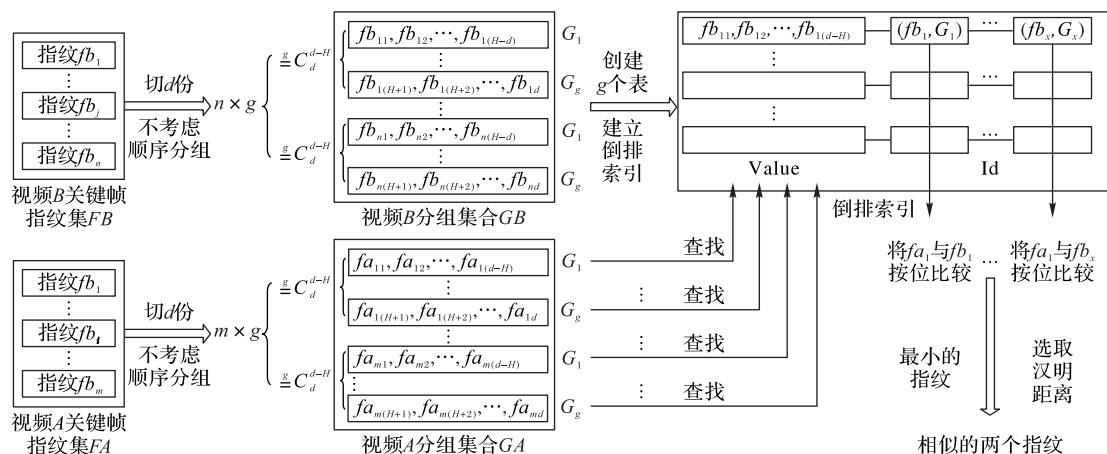


图1 基于切块的快速对比过程

Fig. 1 Fast comparison process based on dicing

假设按位对比两个指纹所用时间为 $\tau$ ,在切分块数为 $d$ 、汉明距离为 $H(H < d)$ 的已知条件下,对比 $N$ 个指纹所需的大致时间 $T$ 如式(1)所示:

$$T = N \times \frac{g}{2^d} \times \tau \quad (1)$$

$$g = C_d^{d-H} \quad (2)$$

其中:  $C_d^{d-H}$  代表随机在  $d$  中不放回地选取  $d-H$  个有  $g$  种可能方式。

需要注意的是,由于指纹在进行切块操作后,形成的是数量不定的不同值,即生成的倒排索引的关键词以及文档的数量也不定,因此所需时间并不是简单的关于时间的线性函数,式(1)仅仅是求出对比  $N$  个指纹所需的大致时间。

可以看出,在  $d < 2$  的情况下,  $T < N \times \tau$ , 即能达到比时间缩短的效果。

## 2 基于切块的相似视频检测方法

基于感知哈希和切块的相似视频检测方法主要包含关键帧提取单元、指纹集生成单元、指纹切块单元、倒排索引建立单元、指纹对比单元以及相似度判定单元。

基于感知哈希和切块的视频比对方法流程如图 2 所示。首先利用帧间差分算法来实现视频中的关键帧提取,再利用感知哈希算法来提取关键帧中的指纹,最后通过切块的思想对比指纹,从而计算两个图像帧的相似度可以在图像对比方面判定所给的视频是否为相似视频。

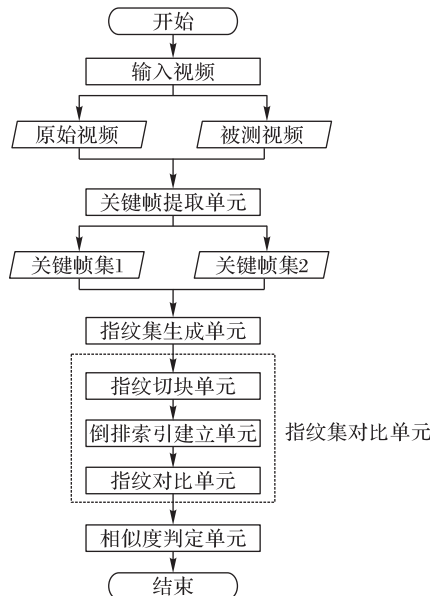


图 2 基于感知哈希和切块的视频对比方法流程  
Fig. 2 Flowchart of video comparison method based on perceptual hashing and dicing

### 2.1 关键帧提取单元

关键帧指的是角色或者物体运动或变化中的关键动作所处的那一帧。高效的视频关键帧提取算法能有效简化处理视频对比问题的计算量。本文采用基于帧间差分的关键帧提取算法<sup>[20]</sup>对视频的关键帧进行提取。

基于帧间差分的关键帧提取算法流程如算法 1 伪代码所示:首先,将所需要对比的视频进行切帧处理得到一个图像帧的集合;接着对集合中相邻两帧的图像进行差分计算,得到平均像素强度;再寻找并提取平均帧间差分强度的局部最大值的图像帧作为视频的关键帧。

算法 1 基于帧间差分的关键帧提取算法。

输入 视频  $V$ , 阈值  $TH$ ;

输出 关键帧集  $KF$ 。

- 1) 帧集  $VS \leftarrow$  将视频  $V$  进行切帧处理;
- 2) 计数  $count \leftarrow 0$ ;

- 3) for  $i \leftarrow 0$  to  $len(VS)-1$  do
- 4) 帧间差  $diff \leftarrow VS[i+1] - VS[i]$ ;
- 5) if  $diff > TH$  do
- 6) 候选帧集  $cds[count] \leftarrow VS[i]$ ;
- 7) end if
- 8) end for
- 9) 初始化关键帧集  $KF$ ;
- 10) 计数  $count \leftarrow 0$ ;
- 11) for  $i \leftarrow 0$  to  $len(cds)$  do
- 12)  $diff\_max \leftarrow$  找出每隔  $l$  帧的局部最大差值;
- 13) if  $diff\_max > TH$  do
- 14)  $KF[count++] \leftarrow diff\_max$ ;
- 15) end if
- 16) end for

### 2.2 基于感知哈希的指纹集生成单元

指纹集生成单元的作用是生成能够代表视频的指纹集,该单元的核心就是利用感知哈希算法,通过 DCT 将图像从像素域转换为频域,只保留系数矩阵的左上角区域的元素用来计算能够代表图像的哈希值。DCT 计算如式(3)所示:

$$\begin{cases} D(u, v) = c(u)c(v) \sum_{i=0}^{X-1} \sum_{j=0}^{Y-1} e(i, j) \theta(i, j) \\ \theta(i, j) = \cos\left[\frac{(i+0.5)\pi}{X} u\right] \cos\left[\frac{(j+0.5)\pi}{Y} v\right] \\ c(u) = \begin{cases} \sqrt{1/Y}, & u = 0 \\ \sqrt{2/Y}, & u \neq 0 \end{cases} \end{cases} \quad (3)$$

式中:  $(u, v)$  代表着像素点的空间位置,  $e(i, j)$  是输入图像的像素点,  $X$  和  $Y$  是  $e$  的行数和列数。

详细的感知哈希算法流程如算法 2 伪代码所示。为了保证压缩的图像在保留大多数信息的前提下,简化 DCT 的计算过程,提高指纹生成速度,本文将图片尺寸缩小到  $32 \times 32$ 。得到 DCT 矩阵后,选取左上角  $8 \times 8$  最能代表图像整体信息的低频成分,通过与 DCT 均值比较后,得到最终的 64 位图像指纹序列。

算法 2 基于感知哈希的图像指纹算法。

输入 图像  $G$ ;

输出 数字指纹  $f$ 。

- 1)  $G_1 \leftarrow$  将图像  $G$  转换为灰度图;
- 2)  $D_1 \leftarrow$  计算灰度图  $G_1$  的 DCT 系数矩阵;
- 3)  $D_2 \leftarrow$  只获取左上角  $lf \times lf$  的系数矩阵;
- 4)  $DCT\_avg \leftarrow$  计算  $D_2$  均值;
- 5) 计数  $count \leftarrow 0$ ;
- 6) for  $v$  in  $D_2$  do
- 7) if  $v < DCT\_avg$  do
- 8)  $f[count++] \leftarrow 0$ ;
- 9) else do
- 10)  $f[count++] \leftarrow 1$ ;
- 11) end if
- 12) end for

通过 pHash 算法获得的哈希值只能粗略地获得与平均频率的相对比率,而不能获得真正的低频分量。然而,只要图像的大致结构不被改变,通过该算法获得的指纹就不会改变,因此该算法可以避免颜色直方图和伽马校正的影响。

### 2.3 指纹切块及倒排索引建立单元

指纹切块单元将感知哈希得到的指纹进行切块操作,平均切成  $d$  个等长的块,为后续建立倒排索引做好准备。在确定了汉明距离为  $H$  的情况下,对切分后的  $d$  个指纹块随机取

出  $d-H$  块, 则共有  $g$  个组合方式, 计算方法如式(2)所示。

根据倒排索引的结构, 针对某一组合方式中的指纹块的值, 倒排索引能快速找到包含该值的指纹编号, 即对应的若干指纹有相似的可能。因此得到切块之后, 对这  $g$  个指纹块形成的值构建倒排索引。

#### 2.4 指纹对比及相似度判定单元

指纹对比单元将遍历倒排索引中的元组, 找到包含某值的完整指纹, 按位比较该值对应的指纹。基于切块的对比算法如算法3伪代码所示。

算法3 基于切块的对比算法。

输入 指纹集  $FA$  和  $FB$ , 切块数  $d$ , 汉明距离  $H$ ;

输出 视频相似度  $S_{FA, FB}$ 。

- 1) 将指纹集  $FA$  中的指纹切为  $d$  块;
- 2) 将指纹集  $FB$  中的指纹切为  $d$  块;
- 3)  $FB \leftarrow$  指纹数量多的指纹集,  $FA \leftarrow$  指纹数量少的指纹集;
- 4) 倒排索引  $Inv \leftarrow$  对指纹集  $FB$  建立倒排索引;
- 5) 计数  $count \leftarrow 0$ ;
- 6) for  $i \leftarrow 0$  to  $d \times \text{len}(FA)$  do
- 7)  $fa \leftarrow$  找到该指纹块对应  $FA$  集的完整指纹  $FA(i/d)$ ;
- 8)  $fb \leftarrow$  通过倒排索引在  $FB$  集中找到包含该指纹段的完整指纹  $Inv(fa(i))$ ;
- 9) if  $\text{hamming}(fa, fb) < H$  do
- 10)  $count \leftarrow count + 1$ ;
- 11) end if
- 12) end for
- 13) 计算指纹相似度  $S_{FA, FB} \leftarrow \frac{count}{\min(\text{len}(FA), \text{len}(FB))}$ ;

相似度判定单元中将判定两指纹的汉明距离是否小于阈值  $H$  (本文认为汉明距离不大于  $H$  的两帧图像是相似的), 得到两个视频相似指纹的集合, 最后得出两个视频的图像相似度。指纹相似度的计算方式如下:

$$S_{A, B} = \frac{s_{\text{sum}}}{\min(n, m)} \quad (4)$$

式中:  $s_{\text{sum}}$  表示原始视频和被测视频中相似指纹个数,  $n$  和  $m$  分别表示原始视频  $A$  指纹和被测视频  $B$  指纹的个数。

### 3 实验结果与分析

为了验证基于切块的相似视频检测方法的有效性, 本章将在作者构建的广告数据集上与传统感知哈希的相似视频检测方法进行对比, 以验证速度的提升; 并在公开数据集 CC\_WEB\_VIDEO<sup>[17]</sup> 上与其他三种主流方法进行准确度和速度的对比。

作者构建的广告数据集中的原始视频主要是由从腾讯视频等视频网站上下载的6个时长为15~77 s的广告视频, 通过对其添加滤镜、乱序剪辑、添加水印及抽帧等常见视频操作后形成被测视频。

CC\_WEB\_VIDEO 数据集由12 790个时长为5 s到10 min的共800 h的视频片段组成, 这些视频是从YouTube等视频分享网站上下载的。根据搜索的关键词分类, 该数据集分为24类。在每一类中, 将最流行的视频作为原始视频, 其他视频作为被测视频, 平均有27%的视频与原始视频重复或近似重复, 且每个重复版本都是用户自行上传, 没有经过额外的人为操作。

本文将根据文献[21]中提到的查准率  $P$ 、查全率  $R$  以及平均准确率均值 (mean Average Precision, mAP) 对本文所提

方法进行评估:

$$P = TP / (TP + FP) \quad (5)$$

$$R = TP / (TP + FN) \quad (6)$$

$$I_{\text{mAP}} = \int_0^1 P(R) dR \quad (7)$$

其中:  $TP$  代表正确检测到的相关视频数量,  $FP$  代表错检视频数量,  $FN$  是漏检视频数量。

#### 3.1 预处理

实验首先针对原始视频和被测视频进行关键帧提取预处理, 以降低后续比较过程的时间复杂度。

下面分别以名为“1. mp4”、长为15 s的原始视频, 以及添加滤镜及乱序之后名为“1\_1. mp4”、长仍为15 s的被测视频为例, 通过OpenCV中的VideoCaptrue函数对视频进行逐帧读取, 会在内存中得到一个关于该视频的所有图像帧的集合。接着对图像帧集合按照前后两帧的顺序依次进行差分计算, 得到一个平均像素强度的集合; 与本文给定的阈值0.6进行对比, 得到一个候选的关键帧集合。最后通过查找平均帧间差分强度的局部最大值确定视频的关键帧。图3和图4分别是根据查询视频和被测视频形成的关键帧。



图3 “1. mp4”提取的关键帧

Fig. 3 Key frames extracted from “1. mp4”



图4 “1\_1. mp4”提取的关键帧

Fig. 4 Key frames extracted from “1\_1. mp4”

原始视频所截取的关键帧数为21, 而被测视频所截取的帧数为25, 由此可以看出相比原始视频中375的帧数, 为后续的相似帧检测降低了时间复杂度。而两个视频提取的关键帧数量不同, 是因为被测视频对原始视频进行了乱序操作, 改变了帧的顺序, 因此在计算相邻帧间的差分值时存在差异, 也就导致两个视频最后得到的帧数量不同。

#### 3.2 视频关键帧指纹生成

本节将对上述的关键帧集合做感知哈希算法以提取关键帧指纹集, 用于后续的相似度判定。

##### 3.2.1 感知哈希算法鲁棒性验证实验

除本文使用的感知哈希算法外, 图像指纹算法最常见的还有均值哈希算法 (average Hashing, aHash) 和差值哈希算法 (difference Hashing, dHash)。均值哈希是将压缩后的图像的像素点与所有像素的均值做大小比较的二值化处理, 得到64

位指纹;差值哈希是将压缩后的图像的像素点与相邻像素点做差值计算后进行二值化处理,得到 64 位的指纹。将感知哈希与均值哈希、差分哈希两种常用哈希算法进行鲁棒性验证对比实验。

对图 5(a)所示的第 222 帧图片进行添加滤镜、更改尺寸、镜像、伽马校正以及更改颜色直方图共五种操作。在 6 块

Intel Xeon CPU 配置情况下,得到结果如表 1 所示,由表中数据分析可得,虽然感知哈希的耗时明显大于均值哈希和差分哈希,但是在五种操作(特别是在添加滤镜、更改尺寸和镜像)下,相似度都高于均值哈希和差分哈希。因此,综合视频处理最常见的五种操作考虑,本文选取鲁棒性较强的感知哈希算法作为指纹集生成算法。

表 1 各算法在不同视频处理操作下的性能对比

Tab. 1 Performance comparison of different algorithms under different video processing operations

| 算法    | 添加滤镜  |        | 更改尺寸  |        | 镜像    |        | 伽马校正  |        | 更改颜色直方图 |        |
|-------|-------|--------|-------|--------|-------|--------|-------|--------|---------|--------|
|       | 相似度/% | 耗时/ms  | 相似度/% | 耗时/ms  | 相似度/% | 耗时/ms  | 相似度/% | 耗时/ms  | 相似度/%   | 耗时/ms  |
| aHash | 95.3  | 0.798  | 98.4  | 0.825  | 40.6  | 0.778  | 87.5  | 0.791  | 100.0   | 0.827  |
| dHash | 100.0 | 0.171  | 95.3  | 0.162  | 59.4  | 0.162  | 98.4  | 0.162  | 95.3    | 0.189  |
| pHash | 99.2  | 34.540 | 99.6  | 21.720 | 89.1  | 33.010 | 98.4  | 41.870 | 99.6    | 30.730 |



图 5 视频处理的五种常见操作

Fig. 5 Five common operations of video processing

### 3.2.2 指纹生成

得到上述的关键帧集合后,通过感知哈希算法分别得到两个视频关键帧的指纹。

以图 3 中所示的原始视频中的第 222 帧与图 4 中所示的被测视频中的第 93 帧为例,首先将得到的关键帧进行尺寸缩小,在这里为了简化 DCT 运算过程,将图像缩小为  $32 \times 32$ 。接着计算该图像帧的 DCT,得到一个  $32 \times 32$  的 DCT 系数矩阵图。得到该 DCT 系数矩阵之后,本文只保留在左上角的  $8 \times 8$  大小的矩阵,因为它显示了这张图像的最低频成分。然后对所有的关键帧进行如上的步骤,并且计算 DCT 的均值。将根据 pHash 算法得到的 64 个 0 或 1 的数字组成一个 64 位的整数,即得到帧 222 与帧 93 的一组指纹,如表 2 所示。得到需要对比的关键帧的指纹之后,经计算两帧的汉明距离为 2,在阈值 7 之内,证明这两帧的关键帧是相似的。

表 2 帧 222 与帧 93 的指纹序列

Tab. 2 Fingerprint sequences of frame 222 and frame 93

| 关键帧   | 指纹序列                                         |
|-------|----------------------------------------------|
| 帧 222 | 10111100111010000110101000110010000100100000 |
|       | 01100010100010010101                         |
| 帧 93  | 10111100111010000110101000110010000100100000 |
|       | 01100010100010010101                         |

### 3.3 视频相似度对比结果及分析

为了验证基于切块的相似视频检测方法的有效性,将本方法与传统的基于感知哈希的方法在从网上下载的 6 个广告上进行对比,以验证本文方法在速度上的表现;对其他三个常见方法在公开数据集 CC\_WEB\_VIDEO 上进行了对比实验,以验证本文方法在准确度以及速度上的性能。

#### 3.3.1 与传统的基于感知哈希的方法进行对比

由于图像指纹长度为 64 位,为了不丢失图像信息,指纹长度需要能被切块数整除,即切块数应为 64 的因数(即  $d$  取值

应为 2、4、6、8、16、32)。根据经验,汉明距离  $H \leq 5$  的两张图片为相似度极高的图片。并且根据式(1), $d$  和  $H$  越接近,对比时间越短,因此为了保证对比时间并且能够保证检测到大多数相似图像,综合考虑,本实验采用的切块数  $d = 8$ ,汉明距离  $H = 7$ 。根据 6 组现有广告得到图 6 的实验结果,可以看出,本文所提出的基于切块的视频相似度检测方法传统的感知哈希对比方法相比,能够在准确度一致的前提下,缩短约 93% 的检测时间。

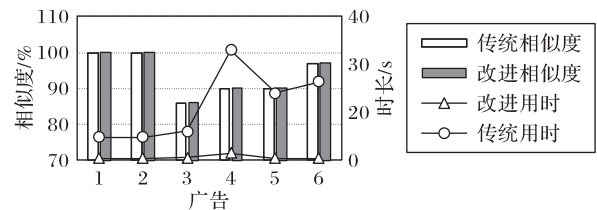


图 6 本文方法与传统方法在广告数据集上的性能对比

Fig. 6 Performance comparison of the proposed method and traditional method on advertising dataset

总体上,感知哈希算法能够较好地实现视频对比的功能,但是可以看到广告 3 不管是在传统算法还是改进算法中的相似度都较低,这是因为感知哈希在改变图像锐度、对比度时检测效果不够优良,因此后续还需进一步寻找更优良的指纹生成算法,以适应更多场景下的视频相似度检测。

#### 3.3.2 与其他方法进行对比

本节将在 CC\_WEB\_VIDEO 数据集上的对所有的视频(包含 24 类)进行实验,并将本文方法与在引言部分提到的多特征哈希(MFH)<sup>[12]</sup>、自学习哈希(STH)<sup>[13]</sup>以及光哈希(SPH)<sup>[14]</sup>三种常见的相似视频检测方法进行对比。

从图 7 可以看出,相比其他三种常见的方法,本文方法在 mAP 上有所提高,分别提高了 1.4%、2% 和 2.3%;而相较于 MFH 检测平均每一个视频耗时 0.553 s、STH 耗时 0.623 s、SPH 耗时 0.496 s,本文方法仅用了 0.418 s,检测时间分别缩短了 24%、32% 和 16%,提高了检测效率。与采用局部特征的对比方法相比,基于切块的视频相似度检测方法是通过对计算关键图像帧之间的相似度来计算视频的相似度,因此更能考虑到绝大多数图像帧,从而能够比上述三种方法的平均准确度有所提高;并且由于优化了传统的遍历对比方法,因此检测时间比其他三种常见方法也有所减短,达到了高效检测相似视频的效果。

本文实验代码和广告数据集已上传 GitHub ([https://github.com/jiangtaoluo/video\\_sim\\_detect](https://github.com/jiangtaoluo/video_sim_detect))。

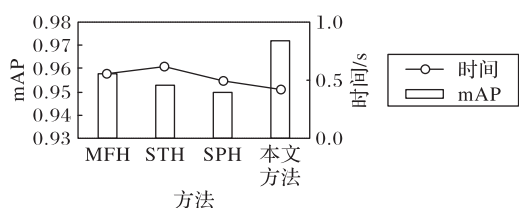


图7 各方法在CC\_WEB\_VIDEO数据集上的性能对比

Fig. 7 Performance comparison of different methods on CC\_WEB\_VIDEO dataset

#### 4 结语

针对现有的视频相似度检测技术需求较多但识别效果局限等问题,本文提出了一种基于感知哈希和切块的视频相似度快速检测方法。本文首先通过基于帧间差分的方法提取视频关键帧,再利用感知哈希算法对比提取出来的关键帧,基于切块数量和汉明距离之间的关系构建倒排索引,从而快速得到两个视频图像帧的相似度。最终在保证准确度的前提下,达到快速检测相似视频的目标。但是由于感知哈希在改变图像锐度、对比度时检测效果不够优良,而前期生成指纹的算法的准确度对后续对比有着显著影响,因此寻找一种优良的指纹生成算法将是需要进一步研究的内容。

#### 参考文献 (References)

- [1] APARNA J R, AYYAPPAN S. Image watermarking scheme based on MD5 and simplified DES [C]// Proceedings of the 1st International Conference on Computational Systems and Communications. Piscataway: IEEE, 2014: 181-186.
- [2] ZHAO M, YAGNIK J, ADAM H, et al. Large scale learning and recognition of faces in web videos [C]// Proceedings of the 8th IEEE International Conference on Automatic Face and Gesture Recognition. Piscataway: IEEE, 2008: 1-7.
- [3] 彭雨蒙. 基于序列比对的视频指纹检索算法研究[D]. 成都:电子科技大学, 2015:19. (PENG Y M. Research on video fingerprint retrieval algorithm based on sequence alignment [D]. Chengdu: University of Electronic Science and Technology of China, 2015:19)
- [4] SUN F, SHI M, LIN W. Feature label extraction of online video [C]// Proceedings of the 2012 International Conference on Computer Science and Electronics Engineering. Piscataway: IEEE, 2012: 211-214.
- [5] BELKHATIR M, TAHAYNA B. Near-duplicate video detection featuring coupled temporal and perceptual visual structures and logical inference based matching [J]. Information Processing and Management, 2012, 48(3): 489-501.
- [6] LONG C, BASHARAT A, HOOGS A. A coarse-to-fine deep convolutional neural network framework for frame duplication detection and localization in forged videos [C]// Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. Piscataway: IEEE, 2019: 1-10.
- [7] SHEN L, HONG R, ZHANG H, et al. Video retrieval with similarity-preserving deep temporal hashing [J]. ACM Transactions on Multimedia Computing, Communications, and Applications, 2019, 15(4): 1-16.
- [8] JIA S, XU Z, WANG H, et al. Coarse-to-fine copy-move forgery detection for video forensics [J]. IEEE Access, 2018, 6: 25323-25335.
- [9] SINGH G, SINGH K. Video frame and region duplication forgery detection based on correlation coefficient and coefficient of variation [J]. Multimedia Tools and Applications, 2019, 78(9): 11527-11562.
- [10] REVAUD J, DOUZE M, SCHMID C, et al. Event retrieval in large video collections with circulant temporal encoding [C]// Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2013: 2459-2466.
- [11] SONG J, YANG Y, HUANG Z, et al. Effective multiple feature hashing for large-scale near-duplicate video retrieval [J]. IEEE Transactions on Multimedia, 2013, 15(8): 1997-2008.
- [12] ZHANG D, WANG J, CAI D, et al. Self-taught hashing for fast similarity search [C]// Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM, 2010: 18-25.
- [13] YELA D F, STOWELL D, SANDLER M. Spectral visibility graphs: application to similarity of harmonic signals [C]// Proceedings of 27th European Signal Processing Conference. Piscataway: IEEE, 2019: 1-5.
- [14] ZHU L, HUNAG Z, LIU X, et al. Discrete multimodal hashing with canonical views for robust mobile landmark search [J]. IEEE Transactions on Multimedia, 2017, 19(9): 2066-2079.
- [15] 石慧杰. 基于卷积神经网络和频率域特征的视频拷贝检测方法 [J]. 高技术通讯, 2019, 29(12): 1201-1205. (SHI H J. Video copy detection based on deep learning features and DCT coefficients [J]. Chinese High Technology Letters, 2019, 29(12): 1201-1205.)
- [16] YAN H, XU X. End-to-end video subtitle recognition via a deep residual neural network [J]. Pattern Recognition Letters, 2020, 131: 368-375.
- [17] WU X, HAUPTMANN A G, NGO C W. Practical elimination of near-duplicates from web video search [C]// Proceedings of the 15th ACM International Conference on Multimedia. New York: ACM, 2007: 218-227.
- [18] 杨文娟,王文明,王金玉,等. 基于感知哈希和视觉词袋模型的图像检索方法 [J]. 图学学报, 2019, 40(3): 519-524. (YANG W J, WANG W M, WANG Q Y, et al. Image retrieval method based on perceptual hash algorithm and bag of visual words [J]. Journal of Graphics, 2019, 40(3): 519-524.)
- [19] 蒋洪. 关于鸽巢原理和Ramsey定理的几个结论 [J]. 科教文汇, 2008(32): 281. (JIANG H. Several conclusions on pigeonhole principle and Ramsey theorem [J]. The Science Education Article Collects, 2008(32): 281)
- [20] LUO J, PAPIN C, COSTELLO K. Towards extracting semantically meaningful key frames from personal video clips: from humans to computers [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2009, 19(2): 289-301.
- [21] TAN H K, NGO C W, CHUA T S. Efficient mining of multiple partial near-duplicate alignments by temporal network [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2010, 20(11): 1486-1498.

**WU Yue**, born in 1997, M. S. candidate. Her research interests include computer vision.

**LUO Jiangtao**, born in 1971, Ph. D., professor. His research interests include mobile big data, next-generation network technology, communication network testing and optimization.

**LIU Rui**, born in 1995, M. S. His research interests include computer vision.

**HU Zhongyin**, born in 1999. Her research interests include computer vision.