華南理工大學
South China University of Technology

《图像处理基础》 课 程 论 文

(2019-2020 学年第一学期)

视频版权检测算法

学生姓名：宋雨杭

提交日期：2019 年 12 月 18 日　　学生签名：

学　　号	201736721010	座位编号	无
学　　院	软件学院	专业班级	软件工程 4 班
课程名称	图像处理基础	任课教师	吴秋霞
教师评语：			
本论文成绩评定： _____分			

说　明

1、课程论文一般要有题目、作者姓名、摘要、关键词、正文及参考文献。

2、论文要求自己动手撰写，如发现论文是从网上下载的，或者是抄袭剽窃别人文章的，按作弊处理，本门课程考核成绩计 0 分。

3、课程论文用 **A4** 纸双面打印。字体全部用**宋体简体，题目**要求用**三号字加粗，标题行**要求用**小四号字加粗，正文**内容要求用**小四号字；**英文论文字体全部用 **Times New Roman，题目**要求用 **18 号字加粗**；**标题行**要求用 **14 号字加粗，正文**内容要求用 **12 号字；**行距为固定值 20 磅；页边距左为 2.5cm、右为 2.5cm、上为 2.5cm、下为 2.5cm；其它格式请参照本科设计（论文）的要求。

4、论文选题、篇幅、内容等由任课教师提出具体要求。

视频版权检测算法

宋雨杭

摘要：本文提出了一种基于特征提取的视频版权检测算法，分别采用 SIFT、CNN、pHash、直方图提取特征进行近似匹配，对比效果优劣，并设计出了一种可用于生产环境的视频检测系统。

关键词：短视频；版权检测；近似图像匹配；图像特征提取；SIFT；CNN；预训练模型；感知哈希算法；直方图匹配；FAISS；

一、引言

近年来，以抖音、快手为代表的短视频平台快速崛起，短视频已形成一个完整的产业链，短视频行业快速发展，近三年内市场规模增长超过 700%（2019 年短视频行业研究报告）。但在快速发展的过程中，短视频产业也面临着诸多问题，其中就有短视频侵权问题就广为诟病。有不少短视频作者选取电影、电视剧、综艺节目等作为片源，然后通过剪辑、快放等选取其中的部分片段来制作短视频，而没有为长视频作者付费。这种做法极大的侵害了长视频作者的权益，长此以往将会给长视频创作带来灾难性的影响，我们再也难看到优质的长视频产出。由于短视频处于一个爆发性增长的状态，依靠人工检测是完全不现实。为了保护视频制作公司及原创者权益，需要通过自动化方式进行针对短视频的侵权行为检测。

短视频版权检测面临着诸多挑战。通常情况下，短视频由长视频剪辑而来的过程中，往往伴随着分辨率、帧率的调整；也有可能插入模板，包括标题、logo、水印、小动画等，同时常常混剪多个视频片段，有时还会出现画中画、图像平移、放大、高斯模糊、部分马赛克等变换。这些变换都给视频版权检测带来了诸多的困难。

目前，视频版权算法还尚不成熟，还没有一个很好的方法能够高准确率的检测视频。但视频检测算法极大的依赖于图像近似搜索，在图像近似搜索的领域已有非常丰硕的成果，例如感知哈希算法、角点检测、SIFT 尺度不变特征以及近几年提出的 CNN 深度特征等。我会在后面对比这些算法在视频版权近似检测中表现出的效果。

本文的创新点在于提出了一套完备的视频版权检测流程，基于 SIFT 特征提取和 FAISS 向量数据库引擎。同时本文对比了多种特征提取方式的效果，通过实验验证 SIFT 凭借着旋转、平移、缩放等不变性，在视频版权检测方面效果最好。

二、相关文献综述

视频近似匹配目前还是一个少有人涉足的领域，相关研究文献极少，也难以见到实

际应用。我们可以将视频近似匹配降维看作是图像的近似匹配。理想状态下，只有能将给定的视频提取的帧与参考视频中的帧完美匹配，就能从给定视频中确定抄袭的部分。

而图像近似匹配算法历史悠久，有着丰富的研究成果[6] [17]。1998 年 Chris Harris 提出了角点检测算法[15]，使用一个固定窗口在图像上进行任意方向上的滑动，通过对比滑动后灰度变化的程度，来判断图像中是否存在角点。然后将角点作为关键点，该点附近的灰度变换特征就是特征向量了[14]。2004 年，David G. Lowe 提出了 SIFT 尺度不变特征变换[10]。SIFT 算法通过探测空间极值点，得到关键点，然后对关键点附近的像素生成一个 128 维的特征向量[11]。这种方法一度成为当时最好的图像关键点检测、近似匹配的方法。后来，Herbert Bay 对 SIFT 算法进行了改进，在牺牲了一定准确率的情况下，提出了更快的 SURF 特征提取方法。SURF 的特征检测速度会比 SIFT 快几十倍。SURF 会生成 64 维特征向量，可用于图像近似搜索。近些年来，Fred Weinhaus 提出了基于图像感知哈希的图像近似搜索方法[7]，Massimiliano Patacchiola 提出基于色彩直方图的近似匹配[8]，以及近些年非常火热的基于 CNN 的特征提取方法[4] [5]，且用实践证明 TensorFlow 提供的用于图像场景识别的预训练模型中间层可作为图像的特征向量[16]。

本论文通过对比不同特征提取方法在视频版权检测中的实际效果，选定了效果最好的方法，并编写了一份完备的、可用于视频版权检测的代码。

三、方法

我首先按下面第四节里“实验数据”部分所述，下载好实验数据，按 1fps 的帧率从视频中提取图像，作为数据预处理。（代码./code/preprocess. ipynb）

然后，我分别计算 refer 和 val 中每个视频每一帧的 SIFT 特征向量，并将它们按视频储存。这里会用到 opencv 库 xfeatures2d 中的 sift 库文件。但由于 SIFT 在近几年申请了专利，所以新版的 opencv 库已经移除了该功能，需要安装旧版的 opencv。可以使用 opencv-python 3.4.2.17，opencv-contrib-python 3.4.2.17。然后使用 Faiss 作为向量数据库[21]，将每个视频的所有特征向量加到一个数据库中。按照 Faiss 数据库给出的指南[22]，建立所以时选用参数 PCAR16,IVF1024,SQ8，表示进行主成分分析法降维至 16 维，并在索引中压缩至 8 位。FAISS 中的压缩均为有损压缩，压缩后向量距离的相对关系保持不变，但是具体距离值会发生变化。但这并不会影响我们的实验。通过这种方式建立的特征数据库索引，大小均在 100MB 以内。

完成上述操作后，我们得到了每个长视频的特征数据库。然后我们将每个短视频的特征一次放入每个长视频的特征数据库中，将欧氏距离小于 4000 的判定为相同的特征

点。然后比对每个长视频与其能匹配上的数量，将匹配特征点最多的长视频作为匹配结果。

此时已经能够完成视频匹配。下面缩短匹配时间段，即大致找出侵权时间的片段。我用的二分法，将长视频分为 4 段，找出最相似的一段，然后去掉离它最远的一段。重复这个步骤，直到剩余视频长度不大于短视频的 2 倍时停止。完成后，我们就能将侵权时间段锁定到短视频长度的 2 倍以内了。在生产环境中这样做已经符合预期了。上述三个步骤的代码位于./code/SIFT.ipynb。

由于比赛要求匹配误差不大于 5 秒，所以我又再写了一个精确匹配的程序，文件名是 SIFT_infer_exact_time.ipynb。我采用滑动窗口的方式，对视频进行组帧匹配，若长度为 10 的滑动窗口中有 5 个能匹配上，则认为当前滑动窗口中心为正确的匹配点。这样就能精确到 5 秒之内了。

由于所有代码加起来有上千行，所以难以详细叙述每个细节，但总体思路如上所述。对于 CNN、pHash 等方法，其代码和匹配步骤也非常相似，只有特征提取部分换成了相应的库文件。

四、实验数据

本次实验我使用的是 2019 CCF 大数据与计算智能大赛的视频版权检测算法所提供的评测数据。因为该下载链接需要报名后才能获得，这里我将其下载链接、MD5 均放在附录中便于下载。

数据组成及使用情况：

本数据集的视频数据主要分为 refer 数据集、query 训练集、query 测试集。其中，query 测试集是用于该比赛的线上提交，没有数据标注，并不知道其真实结果，所以在本论文中不予使用。为了便于本次论文研究，我从 refer 数据集中随机选出 60 个长视频，从 query 训练集中随机选出 500 个短视频用于本次研究。

我将下载好的数据存放于./unzipped/refer 和./unzipped/val 中，使用 ffmpeg 按 1fps 的帧率提取视频中的帧。即每一秒提取一帧。

```
ffmpeg -i ../unzipped/refer/{file_name} -vsync 2 -r 1 ../frames/refer/{file_name[:-4]}/%05d000.jpg
```

完成后，分别将提取好的关键正存放于./frames/refer_all，./frames/val_all。然后随机抽取 60 个 refer 和 500 个 val 作为样本，将他们复制到./frames/refer, ./frames/all。接着从数据标注 train.csv 中找出对应数据，将其存放于./grondtruth.csv,用于后面的校验。

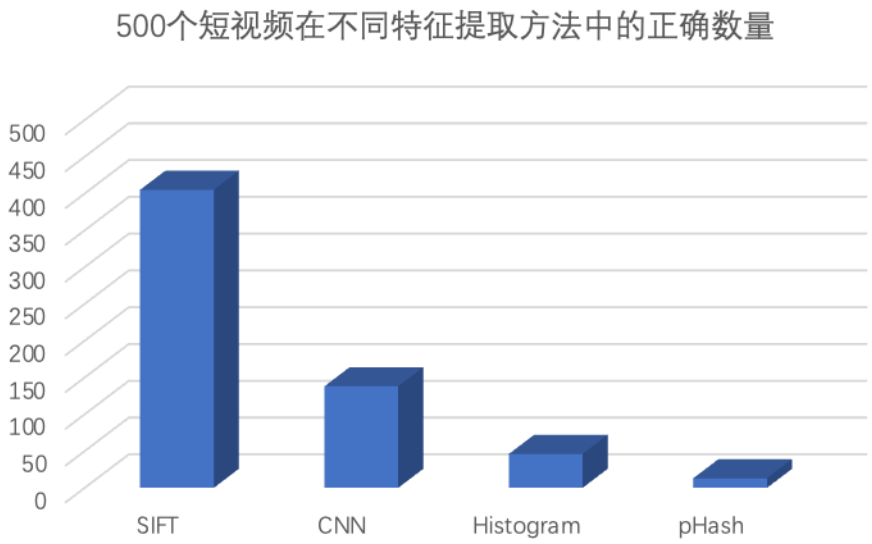
五、实验结果及分析

本次实验中，因为各种方法完成后的效果差别极大，部分效果不好的算法如果按照

CCF 大数据与智能计算竞赛赛题要求的话，得分可能几乎为 0，不便于对比。

于是我重新设计了一下评价标准，首先重新划分了数据集，只对其中随机选出的 500 个视频进行匹配。同时，只考虑是否能够成功匹配到侵权视频，略去侵权时间段精确匹配的操作。

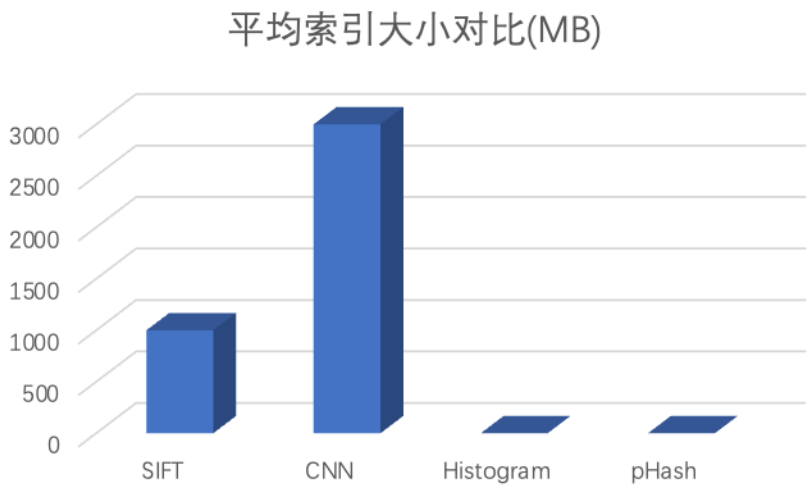
实验数据如下：



500 个短视频在不同特征提取方法中的正确数量

可以看到，直方图匹配和基于感知哈希的匹配因为抗干扰性极差，所以匹配效果很糟糕。CNN 算法因为不具有旋转、缩放等的不变性，所以在面对视频版权检测的时候，效果不如 SIFT 这种传统方法好。具体每种方法的优劣我将在附录里介绍。

实际生产环境中，所提取的特征向量大小也至关重要。因为通常情况下都有海量的视频需要检测版权。下面是一个对比图。SIFT 的特征大小略微比视频本身大，而 CNN 网络（这里用的是 NASNetMobile 预训练网络）提取出的特征大小是视频本身的好几倍。而直方图、感知哈希算法属于传统方法，提取出的特征数据极小。



平均单个长视频特征向量大小的对比，其中 CNN 使用的是 NasNetMobile 预训练网络

对比上述两张图可看出，采用 SIFT 提取特征向量，正确率最高，且提取出的特征占用储存空间与原视频相当尚可接受，对比之下是最好的方法。

附上本论文所给程序在 CCF 大数据与智能计算竞赛中的成绩截图。我们的程序最高能做到 0.44 分（按 F1-score 计算方法计算），即平均而言能够检出 44%的侵权视频并准确匹配出侵权视频所侵权的部分和原视频对应的部分，误差控制在 5s 以内。[18]

六、结论

本论文提出了基于 SIFT 尺度不变特征变换的视频版权检测系统，通过抽取视频帧、计算 SIFT 描述子、建立特征向量索引的方式，实现了侵权视频匹配。然后又使用二分法缩短匹配区间、逐帧比对筛选的方式进行精确匹配，将误差缩小到 5 秒之内。这个计算方法在 CCF 大数据与人工智能竞赛的测试集上做到了 0.44 分，排名 13。同时，我还对比了使用图像感知算法（pHash）、CNN 深度特征提取、直方图比对等方式，发现凭借着旋转、平移、缩放的不变性，在视频版权检测中表现最佳。

未来研究可以继续在 SIFT 等特征点检测算法的基础上，改进匹配的过程，以做到更高的准确率。同时，还能对特征提取的思路进行改进，例如 André Araujo 有在最新研究结果中提到将视频拆分为很多段剪辑的组合，连续相似的画面为一个剪辑，然后对每个剪辑算出一个特征向量来表示。这种方式能有效避免视频过长时被误匹配的情况[19]。

抖音也给出了一个用于短视频近似匹配研究的数据库[19]，相信今后会有更多关于视频版权检测的研究成功。

七、附录

1. SIFT 特征提取

SIFT 全称为 Scale-invariant feature transform，是一种检测局部特征算法，它先在图像中寻找多个关键点。通过检测 DOG 尺度空间极值点的方式找到关键点，然后对于每一个特征点都会生成一个 128 维的关键点描述子，这样就得到了 SIFT 特征。

SIFT 提取出的特征有着非常优良的尺度不变性、旋转不变性。因为其特征是针对一个个特征点的，而不是全局的。这样图片经过任意缩放、剪裁、选装，甚至略微改变图像的拍摄角度，只要这些特征点还在图中，就能被匹配上。所以 SIFT 特征提取的方法表现出的稳健性超出想象。

但另一方面，由于 SIFT 基于的是特征点提取，因此每一帧所提取到的特征点数量并不相同。这意味着每张图片的特征向量维度并不相同。这会给近似匹配带来极大的困难。例如，现有 A、B 两个参考的图像，X 为待匹配的图像，且 X 的真实匹配对象应为 A。假设 A 能通过 SIFT 提取出 50 个特征点，而 B 能通过 SIFT 特征提取出 1000 个特征，那么非常有可能造成 X 与 B 的特征点更容易匹配上这种假真（False Positive）现象。

而且，我们进行实验的数据集主要是由综艺节目组成，而这些节目有个很大的特点：很多镜头极为相似。有好多视频里，主持人、评委都是一群人，穿着一样的衣服，坐在同样的背景前面。同一个人，穿同一个衣服，提取出的特征点也是非常相似的，就会导致错误匹配。而且这是 SIFT 的本质所决定的弱点，因为 SIFT 提取的是局部特征，没有一个全局性。

2. CNN 特征提取

本实验中还用到了 CNN 进行特征提取。简单来讲，就是通过预训练的 CNN 模型，去掉其最后的全连接层，获得中间计算出的深度特征向量。

我是用的 Keras 框架是一个基于 Tensorflow 的深度学习框架，它提供了 Xception、VGG16、VGG19、ResNet、InceptionV3、InceptionResNetV2、MobileNet、MobileNetV2、DenseNet、NASNet 这 10 个预训练的模型。 Keras 中记录了这几个模型对比效果如下。

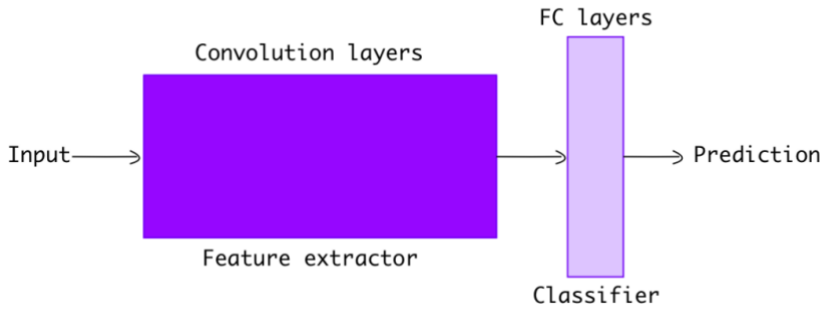
Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	98 MB	0.749	0.921	25,636,712	-
ResNet101	171 MB	0.764	0.928	44,707,176	-
ResNet152	232 MB	0.766	0.931	60,419,944	-
ResNet50V2	98 MB	0.760	0.930	25,613,800	-
ResNet101V2	171 MB	0.772	0.938	44,675,560	-
ResNet152V2	232 MB	0.780	0.942	60,380,648	-
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.750	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
NASNetMobile	23 MB	0.744	0.919	5,326,716	-
NASNetLarge	343 MB	0.825	0.960	88,949,818	-

The top-1 and top-5 accuracy refers to the model's performance on the ImageNet validation dataset.

预训练模型性能的对比

基本上来说，符合特征向量维度越大，准确率越高的特点。尝试对比了不同预训练模型生成的单个长视频特征向量大小后，我选用的是 NASNetMobile.

预训练的图像分类模型的基本原理如下图所示。将图像输入模型后，经过卷积层（convolutional layers）处理，提取出深度特征，具体表现为一个很长的向量。然后再通过全连接层（fully connected layers），对深度特征向量进行识别，从而输出分类。



预训练模型的工作原理

而在本次视频近似匹配的研究中，我们并不需要全连接层来输出分类，因此必须在调用预训练模型的时候指定 include_top=False，排除全连接层，这样输出的直接就是深度特征向量。

由于使用的是预训练好的模型，所以输入的图像大小需要和该模型一致。我尝试了多个模型，其输入大小一般为 224x224，NASNet 和 NASNetMobile 这两个模型要求输入

331x331 大小的图像。而我们视频中提取的图像一般都会大于这个分辨率，因此需要先进行仿射变换，将图像拉伸到指定大小后输入。

使用 CNN 卷积神经网络提取的深度特征有一个天生特性，特征向量的维度是相同的。每张图片都会生成固定维度的向量，这样非常有利于后面的近似检索。这完全解决了前面提到 SIFT 特征向量数不确定的天生缺陷。

然而就如总结中所提到的那样，CNN 提取的深度特征不具有尺度不变性、旋转不变性等特征。也就是说，简单的平移、旋转就会导致得到的深度特征完全不同。

现有的资料描述 CNN 有尺度不变性，是因为其有一个 pooling 的操作，这个操作对局部感受取了极大值。但是这种不变性是极其微弱的，如果进行了放大、缩小、截切、画中画、横向视频转为纵向等等操作后，所获得的深度特征会完全不同。这样即使图像非常相似也难以识别出来。

视频版权检测的根本目的就是要对进行了很大的变化操作的视频进行检测，所以 CNN 深度特征在视频版权检测算法中不适用。

3. **感知哈希算法** Perceptual hash algorithm

感知哈希算法可以对每个图片生成一个值，然后计算两个哈希值的汉明距离（hamming distance），就可以比对两个图片的距离。

感知哈希是一类算法的总称，包括 aHash、pHash、dHash 等。虽然叫做哈希算法，但其与用于加密算法所用的哈希算法有着很大的区别。加密所使用的哈希算法，理想状态下输入数据一个字节的变换都会给最终生成的哈希值带来极大的变换。而图片所用的感知哈希算法则不同，理想的感知哈希算法希望少量图片信息改变不会对哈希值带来太大的变化，并且希望哈希值的变化程度与图片信息改变程度相同。

但运用到视频版权检测中时，和深度特征一样，同样也面临这对于变换后的图像难以识别的问题。同时，如果对画面整体的亮度、色彩对比度进行调整，感知哈希算法也难以得出较好的结果，抗干扰性很差。

由于感知哈希算法主要是对噪声、模糊、压缩等具备较好的不变性，现有的基于感知哈希算法的研究，一般都用于图像失真等领域的研究。例如，定量研究图像经过模拟信号传输后，发生改变的程度。这种情况下感知哈希能够较好的表示出两幅肉眼相同的画面的细微差别程度。而对于我们的要研究的相同场景、有很大差异的图像来说，感知哈希算法不适用。

4. 直方图近似搜索

图像直方图是反映一个图像像素分布的统计表，其实横坐标代表了图像像素的种类，

可以是灰度的，也可以是彩色的。纵坐标代表了每一种颜色值在图像中的像素总数或者占所有像素个数的百分比。

这种方法是计算出图片的直方图，然后对比不同图片的直方图距离，就能找到相似图片。

这种算法优点是对于平移、旋转等会有很好的抗干扰性。不管图像怎么进行剪切重拍，直方图改变都很小。然而直方图匹配有个致命的缺点，在图像整体色调发生改变后，直方图改变巨大。类似的，如果将横向视频上下加上黑边转换为了纵向视频，这时候实际色调因为黑色的加入发生了巨大的改变，直方图也就会发生很大的变化。给视频加上黑边也会有这种情况。

因此，我在研究后认为直方图匹配不适用于视频版权检测。

参考文献

[1] 2019 年短视频行业研究报告，https://zhuanlan.zhihu.com/p/89088828

[2] CCF 大数据与计算智能大赛 https://www.datafountain.cn/special/BDCI2019/talent

[3] 视频版权检测算法比赛 https://www.datafountain.cn/competitions/354/datasets

[4] A Simple Guide to Using Keras Pretrained Models， Shiva Verma, Nov. 12, 2019, https://towardsdatascience.com/step-by-step-guide-to-using-pretrained-models-in-keras-c9097b647b29

[5] Keras Documentation, Applications, https://keras.io/applications/

[6] Image Similarity Comparison, https://cotin.tech/Algorithm/ImageSimilarityComparison/

[7] Tests Of Perceptual Hash (PHASH) Compare Metric, Fred Weinhaus, http://www.fmwconcepts.com/misc_tests/perceptual_hash_test_results_510/index.html

[8] The Simplest Classifier: Histogram Comparison, Massimiliano Patacchiola, Nov 12, 2016, https://mpatacchiola.github.io/blog/2016/11/12/the-simplest-classifier-histogram-intersection.html

[9] Histogram Distance for Similarity Search in Large Time Series, Yicun Ouyang, Feng Zhang, Sept. 2010

[10] Distinctive Image Features from Scale-Invariant Keypoints, David G. Lowe, 2004, https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf

[11] SIFT(Scale-invariant feature transform), Minghao Ning, Apr 9, 2019, https://towardsdatascience.com/sift-scale-invariant-feature-transform-c7233dc60f37

[12] Introduction to SURF (Speeded-Up Robust Features), Deepanshu Tyagi, Mar 20, 2019, https://medium.com/@deepanshut041/introduction-to-surf-speeded-up-robust-features-c7396d6e7c4e

[13] SURF: Speeded Up Robust Features, Herbert Bay, https://www.vision.ee.ethz.ch/~surf/eccv06.pdf

[14] Introduction to Harris Corner Detector, Deepanshu Tyagi, Mar 16, 2019, https://medium.com/@deepanshut041/introduction-to-harris-corner-detector-32a88850b3f6

[15] A Combined Corner And Edge Detector, Chris Harris & Mike Stephens, 1998, http://www.bmva.org/bmvc/1988/avc-88-023.pdf

[16] Identifying Similar Images with TensorFlow, Douglas Duhaime, 28 Aug 2017,

https://douglasduhaime.com/posts/identifying-similar-images-with-tensorflow.html

[17] 移动端图像相似度算法选型，岚依，楚丰, May 13, 2018,
https://www.infoq.cn/article/image-similarity-algorithm-on-mobile-client

[18] CCF BDCI 2019 视频版权检测算法赛题排行榜,
https://www.datafountain.cn/competitions/354/ranking

[19] Large-Scale Video Retrieval Using Image Queries, André Araujo, June 2018,
https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7851077

[20] SVD: A Large-Scale Short Video Dataset for Near-Duplicate Video Retrieval, Qing-YuanJiang

[21] FAISS, FaceBook AI Research, https://github.com/facebookresearch/faiss

[22] Guidelines to choose an index , Matthijs Douze ,
https://github.com/facebookresearch/faiss/wiki/Guidelines-to-choose-an-index

附件 1

数据集下载地址

1. query 测试集（约 5G）

   http://datafountain.int-yt.com/BDCI2019/AiQiYI/test_query.tar.gz

   MD5: adcbf15cc7e422418d2570f30e758d4f

2. query 训练集（约 9.7G）

   http://datafountain.int-yt.com/BDCI2019/AiQiYI/train_query.tar.gz

   MD5: 5660dc9bc3ba28e267dae7726abee80f

3. refer 数据集（约 31.5G）

   http://datafountain.int-yt.com/BDCI2019/AiQiYI/train_refer.tar.gz

   MD5: 73e06fe591c2b9dd3b5c050a1ab6e26a

4. 提交样例

   http://datafountain.int-yt.com/BDCI2019/AiQiYI/submit_example.csv

   MD5: 60c39341fd65f2bcdb3ca98eb2fd5984

5. 训练集标注

   http://datafountain.int-yt.com/BDCI2019/AiQiYI/train.csv

   MD5: 7c19a96f325636b4c291ede9782dadd9