IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 6, Ver. V (Nov.-Dec. 2016), PP 84-86
www.iosrjournals.org

Perceptual Hash Based Video Fingerprinting to Search and Identify Content

A paper by Nikhil Sahni

Abstract: *The goal of this research is to identify a suitable algorithm for searching and identifying content considering the transformations or attacks on a given input and yet be tolerant enough to distinguish between dissimilar files. These transformations can include skew, rotation, contrast adjustment and different compression/formats.*

I. Introduction

With rapid technological advances, videos are more and easily accessible across globe. With this recent growth in video content a new problem has arised i.e content searching and recognition from huge database. Preferred way is to use metadata as the reference however metadata is prone to ambiguity problem therefore fingerprinting is now a better solution.Video fingerprinting is a technique designed to uniquely identify a content by its fingerprint which is generated by a selected process of identifying, extracting and then compressing characteristic component of video. Video fingerprinting is irrespective of watermarking which relies on inserting identifying features into the content itself resulting into changed content, while in case of former content remains same.The major bottleneck for watermarking is that the inserted watermark can be destroyed or distorted during video format encoding/decoding during transmission whereas the fingerprint, aka signature can be done after the distribution.Our problem involves searching and recognition of the content in the given content, therefore we require a algorithm to match the similarity factor between the two whose value ranges from 0-1 i.e 0 as no similarity and 1 being completely similar.

Proposed Solution

Let's suppose we have frame $\mathbf{SF_N}$ of n^{th} second and an array of frames $\mathbf{CF_N[]}$ of a video where size of $\mathbf{CF_N[]}$ i.e. F x D where F is frame per second and D is the duration of the video content to be recognized, also a similarity factor **K** will be used here.

- Since we require a similarity factor which can overcome the attacks over the content which may happen during or after transmission of content, we will use perceptual hashing to uniquely identify frame of the content.
- Input all values of CFN [] and generate hash for the same using RADISH algorithm.
- Now input SFN and generate hash for the same using the RADISH algorithm.
- Now match hash of SFN and values of CFN [] generating a similarity factor K for each comparison.
- Value of K will range between 0-1.

II. Results

Original Image
pHash: **ba19c8ab5fa05a59**

Difference: Greyscale
pHash: **ba19caab5f205a59**
Similarity score: **0.96875**

Difference: Cropped and Increased saturation
pHash: **ba3dcfabbc004a49**
Similarity score: **0.78125**

Difference: Cropped, lower JPEG Quality
pHash: **1b39ccea7d304a59**
Similarity score: **0.8125**

Difference: A different Koala image
pHas: **3d419c23c42eb3db**
Similarity score: **0.5625**

Difference: Image of lady
pHash: **f10773f1cd269246**
Similarity score: **0.5**

As observed value of **K** i.e. **'**similarity score' is higher in first three comparisons even after different attacks, while other results showed significantly less similarity.

III. Conclusion

Perceptual hashing is effective in this case where similarity between two images is to be identified. Implementing this algorithm over the array of frames for a video, we can identify the similarity score for the given content. Potential applications include copyright protection, similarity search for media files, or even digital forensics. For example, YouTube could maintain a database of hashes that have been submitted by the major movie producers of movies to which they hold the copyright. If a user then uploads the same video to YouTube, the hash will be almost identical, and it can be flagged as a possible copyright violation. The audio hash could be used to automatically tag MP3 files with proper ID3 information, while the text hash could be used for plagiarism detection.

References

[1] **https://en.wikipedia.org/wiki/Perceptual_hashing**
[2] **http://blockhash.io/**
[3] **http://elog.io/dev/**
[4] **http://phash.org**
[5] **https://github.com/pragone/jphash**
[6] **http://cloudinary.com/blog/how_to_automatically_identify_similar_images_using_phash**