ViDupe-Duplicate Video Detection as a Service in Cloud

Gousiya Farheen Shaik, Min Chen
Division of Computing and Software Systems, School of STEM
University of Washington Bothell
Bothell, USA
{farheen3, minchen2}@uw.edu

***Abstract*—Detecting near-duplicate videos in the cloud is computationally intensive due to sheer number of videos and their large file sizes. To address this challenge, we propose to improve algorithm efficiency by enhancing and revising existing metadata and content comparison methods, and to enable public access by developing a web service. As a proof of concept, a duplicate video detection system called ViDupe is developed that can quickly scans user's Google drive for duplicates.**

Keywords—duplicate video detection; keyframe extraction; perceptual hashing; audio fingerprint comparison

I. INTRODUCTION

Most cloud drive applications on smartphones and tablets provide auto sync of media content to personal cloud drives. This can lead to multiple copies of near-duplicate videos in the cloud drive. Here, near-duplicate videos often refer to videos whose contents are highly matched with each other but may differ in resolution, brightness, effect (e.g., due to blurring, sharpening, etc.) and video codecs [1][2][3]. Such near-duplicate videos waste storage space, increase processing overhead, and impact user experience. Hence, there is a great need to detect and remove them. In the literature, most existing studies are based on keyframe comparison, i.e., extracting features from videos' characteristic frames (called keyframes) [4], generating signatures and computing their similarities [5].

However, such approach overlooks the importance of audio content in a video. For instance, a video dubbed in English will be considered a duplicate of the same video in French but both versions may be needed for different audience. Our research revises and extends existing duplicate video detection techniques and develops a service called ViDupe to help users optimize cloud storage by detecting duplicate files, highlighting good quality files among duplicates, and suggesting users to act on the duplicates.

II. SYSTEM DESIGN AND IMPLEMENTATION

ViDupe integrates with cloud storage service APIs and run cloud jobs to quickly detect duplicate video files in personal drives. As shown in Fig. 1, ViDupe consists of four modules: frontend module, filter module, PHashGen module, and DeDupe module. Each module runs on a cloud balancer to horizontally scale up or scale down as per the HTTP(S) requests or CPU usage. The duplicates will then be presented to users to act upon. More details about each module are presented below.

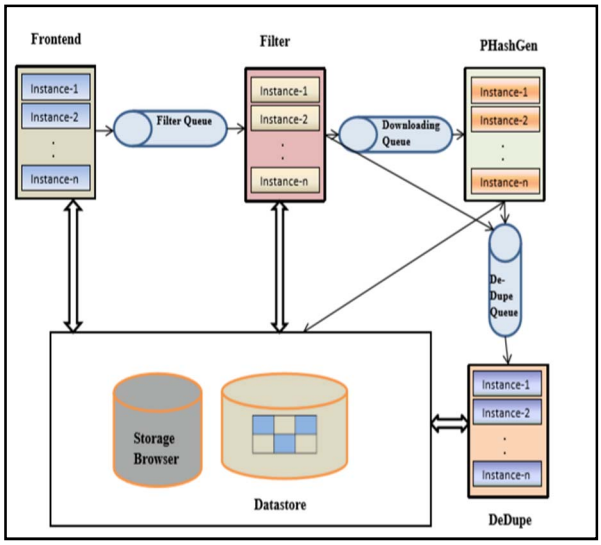

Fig. 1. System architecture

A. Frontend

The Frontend module accepts and forwards users' incoming requests, and index each request as a separate job in the datastore. It accesses a Google API using OAuth 2.0 protocol. Before proceeding, we need to obtain the OAuth 2.0 credentials such as client ID and client secret from the Google API console. The following are the tasks performed by this module.

- Gets an access token form the Google authorization server. This application requires full access to the drive and hence, Google Authorization Server issues the access token based on the access scope.
- Generates a jobID, store user related details and the access token obtained in the Google datastore.
- Sends a message to the Filter module via filter queue, indicating that a new job has arrived and is ready to be processed.

B. Filter

The Filter module aims to improve overall system performance by filtering out videos without duplicates using their metadata. Currently we apply heuristic clustering process on video lengths. The idea is that for a given video database V, if the duration of a video $v_i \in V$ is significantly different from others, a single item cluster $\{v_i\}$ is generated, i.e., v_i has no duplicates and thus can be opted out from further duplicate

978-3-903176-15-7 © 2019 IFIP

searching steps. As heuristic clustering process is a one-pass procedure with complexity of $O(|V|)$, this step can be done efficiently to remove all single item clusters. This results in a set of remaining clusters C, where each cluster $c_i \in C$ contains more than one item, i.e., $|c_i| \geq 2$, to reduce searching space for the *PHashGen* and *DeDupe* modules.

C. PHashGen

PHashGen module generates visual and audio signatures for the remaining videos in clusters C. As discussed in [6], perceptual hashes of near duplicate images are very similar. Therefore, the perceptual hashes for each video keyframe is computed as follows. First each keyframe is converted into grayscale image, on which the Discrete Cosine Transform (DCT) is applied. The average value of DCT coefficients is stored as the perceptual hash value. To create audio signatures, acoustic fingerprinting technology [7] is applied. Audio file is segmented into a set of audio frames whose intensities are computed and their average is stored as its audio fingerprint. In brief, the following steps are performed:

- Extracts keyframes from the video;
- Generates perceptual hashes of the keyframes;
- Extracts the audio file from the video;
- Generates audio fingerprints;
- Writes video and audio hashes to the datastore;
- If all the videos in clusters C are processed, a message is sent to DeDupe module via DeDupe queue;
- Updates the status of the job in the datastore.

As can be seen, *PhashGen* can be computationally intensive even after the filtering process. To address this issue, this module is controlled by load balancers based on the CPU usage to dynamically scale up or scale down the instances as needed. The auto-scaler collects the CPU utilization of the instances in the group and determines whether it needs to scale.

D. DeDupe

DeDupe module is to find the duplicates of each video file. The idea is that for each cluster c_i, keyframes of a video file are compared against the keyframes of the other video files. Two keyframes are said to be similar if the distance between their perceptual hashes lies within a certain threshold value. A similarity measure between two video files is computed by preserving the sequence information of the keyframes. Videos whose visual similarity lie within a certain threshold, undergo audio similarity detection process where the audio fingerprints are analyzed and declared as duplicates if they lie within a threshold limit. In brief, this module executes as follows.

- Fetches all the video and audio hashes of a user, from the Google datastore;
- Checks if the distance between the sets of hashes lie within a threshold;
- Stores the data related to videos as a JSON file in the Google storage;
- Updates the status of the job in the datastore.

After the list of duplicate files are detected, good quality videos with higher resolutions among duplicates are identified and displayed to the user. As shown in Fig. 2, the video with high quality is highlighted by the green box. The application allows the user click each thumbnail image to watch its associated video file and choose any file to delete.

Fig. 2. Screenshot of duplicate video files detected

III. Conclusions

This paper presents a video processing system to help users optimize their drive and curtail their storage cost. It also helps users locate good quality videos in the duplicate set. Future work could include detecting user's duplicate videos across different personal drives like Dropbox, OneDrive etc. In addition, copyright videos are prone to get compromised by unauthorized copying, editing, sharing etc. This often causes huge financial loss to content providers. Invisible digital watermarks are often used to avoid duplication of videos. We can further improve our system to detect duplicates of copyright-protected videos by finding these digital watermarks associated with the videos.

References

[1] H.T. Shen, X. Zhou, Z. Huang, J. Shao, X. Zhou, "Uqlips: A Real-Time Near-Duplicate Video Clip Detection System", *Proc. 33rd Int'l Conf. Very Large Databases (VLDB '07)*, pp. 1374-1377, 2007.

[2] X. Nie, Weizhen Jing, Lin Yuan Ma, Chaoran Cui and Y. Yin, "Two-layer video fingerprinting strategy for near-duplicate video detection," *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, Hong Kong, 2017, pp. 555-560.

[3] S. D. Thepade and A. A. Tonge, "An optimized key frame extraction for detection of near duplicates in content based video retrieval," *2014 International Conference on Communication and Signal Processing*, Melmaruvathur, 2014, pp. 1087-1091.

[4] V. Monga, B. L. Evans, "Perceptual image hashing via feature points: Performance evaluation and tradeoffs", *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3453-3466, Nov. 2006.

[5] J. Meng, H. Wang, J. Yuan and Y. P. Tan, "From Keyframes to Key Objects: Video Summarization by Representative Object Proposal Selection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016, pp. 1039-104.

[6] C. J. Burges, D. Plastina, J. C. Platt, E. Renshaw, H. Malvar, "Using audio fingerprinting for duplicate detection and thumbnail generation", *Proc. Int. Conf. Acoust. Speech Signal Process.*, vol. 3, pp. 9-12, 2005-Mar.

[7] X. Wu, A. G. Hauptmann, and C.-W. Ngo. Practical elimination of near-duplicates from web video search. In Proceedings of the 15th ACM international conference on Multimedia, pages 218–227. ACM, 2007.