# SkipStreaming: Pinpointing User-Perceived Redundancy in Correlated Web Video Streaming through the Lens of Scenes

Wei Liu
Tsinghua University, China

Xinlei Yang
Tsinghua University, China

Zhenhua Li*
Tsinghua University, China

Feng Qian
University of Southern California, USA

## ABSTRACT

When streaming over the web, correlated videos (*e.g.,* a series of TV episodes) appear to bear considerable redundant clips, mostly included in the intros, outros, recaps, and commercial breaks, leading to a waste of network traffic and playback time. Mainstream video content providers have taken various measures to identify these clips, but often result in unexpected and undesirable user experiences. In this paper, we conduct a large-scale, crowdsourced study to demystify the root causes of poor experiences. Driven by the findings, we propose to reconsider the problem from a novel perspective of *scenes* without going through the excessive video frames, which pays special attention to how the contents of correlated videos are organized during video production. To enable this idea, we design efficient approaches to the separation of video scenes and the identification of visual redundancy. We build an open-source system to embody our design, which achieves fast (*e.g.,* taking ~38 seconds to process a 45-minute video using a common commodity server) and accurate (incurring only 770-ms deviation on average) redundancy recognition on representative workloads.

## CCS CONCEPTS

• **Information systems** → **Deduplication**; Web interfaces; • **Computing methodologies** → **Computer vision problems**; **Video segmentation**; Scene understanding.

## KEYWORDS

Correlated videos; web video streaming; visual redundancy detection; video scenes; multimodal (acoustic-narrative-visual) fusion

**ACM Reference Format:**
Wei Liu, Xinlei Yang, Zhenhua Li, and Feng Qian. 2023. SkipStreaming: Pinpointing User-Perceived Redundancy in Correlated Web Video Streaming through the Lens of Scenes. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29–November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3581783.3611845

*Corresponding author.


*MM '23, October 29–November 3, 2023, Ottawa, ON, Canada.*

ACM ISBN 979-8-4007-0108-5/23/10…$15.00
https://doi.org/10.1145/3581783.3611845

## 1 INTRODUCTION

Web video streaming consumes the majority of today's Internet traffic [3, 61], during which correlated videos (*e.g.,* a series of TV episodes or documentaries) are a common type of delivered content [62, 63]. From a user's perspective, correlated videos usually bear considerable redundant clips, mostly included in their intros, outros, recaps, and intermediate commercial breaks. For example, relative to the first episode of *Naruto*, each of the remaining episodes has as long as six minutes of redundant clips, accounting for ~27% of the length of an episode [19]. Naturally, users wish to skip these redundant clips to achieve a more consecutive viewing experience, as well as to save network traffic and playback time [8, 43, 69].

Driven by the wish of users, today's mainstream video content providers (VCPs) like Netflix [10] and Disney+ [6] have provided users with functions for automatically skipping redundant clips during video playback based on pre-generated markers [5, 52]. However, these functions are currently far from satisfactory, resulting in various undesirable experiences and user complaints [2, 9, 12].

To understand why such seemingly easy-to-mark redundant clips could incur poor user experiences, we conduct a large-scale, crowdsourced study on 145,976 correlated videos randomly selected from eight popular VCPs (*i.e.,* Netflix, Amazon Prime Video, Disney+, Hulu, HBO Max, iQIYI, Tencent Video, and Youku). These videos belong to 4,640 series and cover a wide variety of genres like action, anime, comedy, and drama; their information is listed at https://SkipStreaming.github.io. Concretely, we first capture the markers of redundant clips by playing the videos online and meanwhile intercepting the playback information. Then, we recruit 25 volunteers to manually check the correctness of these markers.

Our experiments reveal that a nonnegligible portion (14%) of videos in our dataset have problematic redundancy markers, manifesting as four major symptoms in Table 1. To demystify their root causes, we report them to corresponding VCPs and interview with those who get back to our reports. Also, we survey relevant academic literature to infer the techniques adopted by the remaining VCPs (who do not feed back). Below list our key findings.

- The video and audio encoding schemes adopted by VCPs make it impossible to recognize redundant clips through byte-level comparison. For a series of correlated videos, different encoding parameters are set for different episodes to achieve the optimal balance between quality and bitrate [27, 33, 58]. This makes the raw byte data of visually redundant clips hardly the same.
- Many VCPs use computer vision (CV) techniques, in particular deep learning-based ones [7, 37, 60], to recognize redundant clips. Nevertheless, their performance is constrained by the oftentimes

**Table 1: Common patterns of unexpected and undesired experiences regarding redundant clips in correlated web video streaming.**

| | |
|---|---|
| **Pattern 1: Probabilistic skip on the same clip**<br>A same redundant clip appears in a series of TV episodes. It is automatically skipped during the web-based playback of Episode 3, but not for Episode 7. | All studied VCPs |
| **Pattern 2: Abrupt skip**<br>A redundant clip is skipped but with an incorrect duration that users can perceive. In considerable (16%) cases, the deviation is longer than 5 seconds. | All studied VCPs |
| **Pattern 3: Circumscribed skip**<br>Only the redundant clips at the head or the tail of correlated videos are skipped. | iQIYI, Tencent Video, Youku |
| **Pattern 4: Crash on skip**<br>When a skip action occurs, the web-based video player crashes. | Disney+, Amazon Prime Video |

limited computation resources, especially for the top VCPs that publish tens of thousands of high-resolution videos every day [23, 31]. To mitigate this, even top VCPs would reduce the complexity of CV models and only examine the heading and tailing parts of a video, which inevitably sacrifices the detection accuracy.

- Alternatively, other VCPs compare the audio fingerprints (*i.e.,* spectral features of rendered audios) of correlated videos [4, 73] to detect redundant clips. In practice, this approach is vulnerable to fingerprint collisions and lossy audio encoding. Thus, it is hard to determine suitable similarity thresholds for different sets of videos, causing the detection accuracy to be unpredictable.

To sum up, existing CV-based approaches could be accurate but are rather time-consuming, since they need to analyze typically tens of thousands of video frames when dealing with a correlated video. In comparison, audio fingerprinting-based ones are lightweight but oftentimes inaccurate, because the encoding and fingerprinting processes introduce non-trivial noises and information loss.

In this paper, we reconsider the problem from a novel perspective, *i.e., how the contents of correlated videos are organized during video production.* We note that to make the story coherent and intriguing, today's correlated videos are pervasively organized as a sequence of *scenes.* Representing a series of coherent events that take place in order, *scene* is a widely-used concept in film making [40, 66] to help video editors create more fascinating contents. As exemplified in Figure 1, frames (a)–(e) depict a scene of the intro, and the remainder belong to another scene showing a woman's sweeping the floor.

Further dissecting the correlated videos, we have a key observation that all redundant clips are made up of one or more complete scenes, rather than begin or end arbitrarily within a certain scene. This is quite understandable—a redundant clip (*e.g.,* an intro, an outro, a recap, or a commercial break) is inserted into the main story as a content-wise independent segment during video production. Consequently, if we can effectively locate the scene transitions, we are able to efficiently identify redundant clips by only examining the few video frames near the transitions, without going through the excessive frames that compose the video.

Prior efforts in detecting scene transitions either perform pixel-level comparison between adjacent video frames [28, 36], or continuously analyze visual features such as contrast, color histogram, and

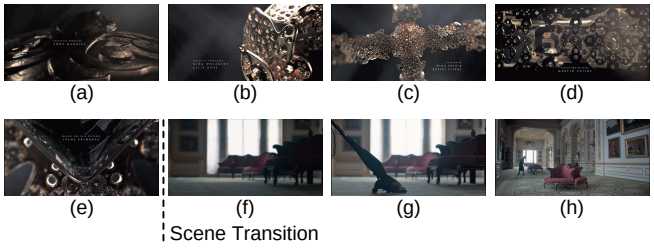


**Figure 1: An example of a *scene* inside the intro sequence of the TV series named *The Crown*, in which a crown is shot from different angles as the background, followed by a scene transition.**

SIFT [41, 56, 66]. However, they all involve intensive data processing on a number of video frames and thus are still heavyweight. To address this, we notice that scene transitions in a video are almost always accompanied by distinct changes in acoustic characteristics of the corresponding audio data. This is because switching to a new scene implies the changes in characters' voices, ambient noises, background music, and so on. For example in Figure 1, the intro scene (frames (a)–(e)) goes with the theme music; when the video moves onto the main story (frames (f)–(h)), the audio part consists of footstep and sweeping sounds. This insight offers us a unique opportunity to efficiently separate scenes, since processing audio data is much more lightweight than processing video contents [34].

To fulfill the above idea, we present *audio-guided scene sketch*, a multimodal (*i.e.,* acoustic-narrative-visual) fusion methodology that uses audio and scene information as crucial guidance to achieve lightweight video redundancy detection. At its core lies our devised novel audio feature named *scene-aware acoustic gradient* (denoted as SceneGrad), which quantifies the variation in acoustic characteristics of audio data to facilitate the detection of scene transitions. Specifically, we first roughly split each video into approximate scenes by searching for the time points that manifest the largest SceneGrad. Then, we perform fine-grained inter-episode video frame matching near these time points to pinpoint the precise boundaries of scenes; to enable high-speed matching, we avoid repetitive frame comparisons by means of *multi-index hashing* [53].

We implement the whole design into an open-source system dubbed SkipStreaming. On our dataset, it achieves a high redundancy recognition precision of 98% and recall of 93% with an average deviation of 770 ms, outperforming existing solutions by over 20%. Note that the 2% false positives are mainly caused by gradual visual transitions such as fades and dissolves, which are not essential to the story and thus would not affect user experiences. In terms of the 7% false negatives, their duration is very short (usually <5 seconds), making them indistinguishable from neighboring scenes and thus hardly affect user experiences either.

For a typical correlated video (*e.g.,* a 1080P, 45-minute TV episode), SkipStreaming takes only 38 seconds to precisely locate redundant clips on a common commodity server with an Intel Xeon 10-core CPU@2.4 GHz, 64-GB memory, and 960-GB SSD. Compared with state-of-the-art (*i.e.,* CV-based) solutions that need tremendous well-annotated video data for training [35, 72], SkipStreaming is faster by at least an order of magnitude without any training prerequisites.

All the source code and data involved in this work are publicly available at https://SkipStreaming.github.io.

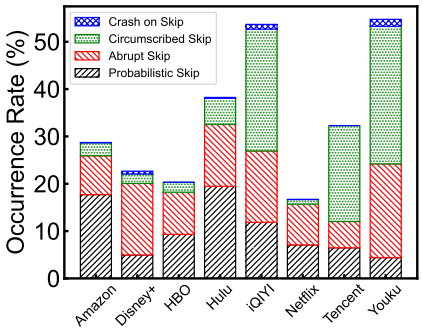


**Figure 2: Classified occurrence rates of unexpected experiences with regard to each VCP.**

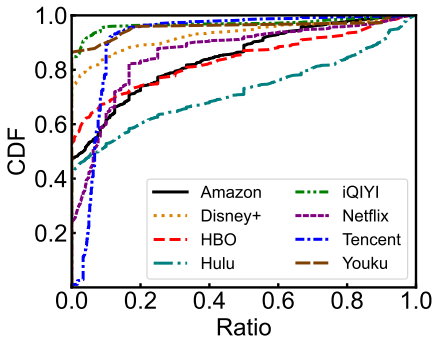


**Figure 3: Ratio distribution of probabilistic skips in the studied series of videos.**

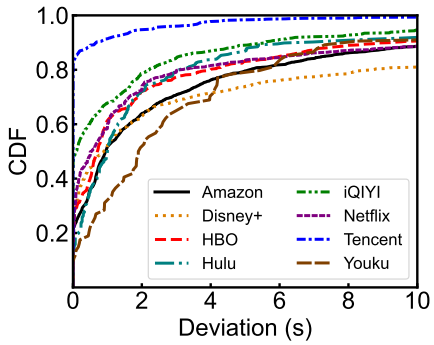


**Figure 4: Distribution of VCPs' marker deviations (set to zero if the marker is correct).**

**Table 2: Statistics of the studied VCPs and correlated videos.**

| VCP | # Genre | # Series | # Episode |
|---|---|---|---|
| Amazon Prime Video | 18 | 711 | 15,495 |
| Disney+ | 17 | 405 | 14,526 |
| HBO Max | 13 | 565 | 17,964 |
| Hulu | 15 | 428 | 22,035 |
| iQIYI | 21 | 478 | 16,024 |
| Netflix | 23 | 865 | 22,758 |
| Tencent Video | 14 | 634 | 18,404 |
| Youku | 20 | 554 | 18,770 |
| All | – | 4,640 | 145,976 |

## 2 MOTIVATION

Watching correlated videos (*e.g.,* a series of TV episodes and shows) has been a prevalent activity among web users, which consumes over 80% Internet traffic in peak hours [3, 49]. On the other side, today's correlated videos usually bear considerable visually redundant clips in intros, outros, recaps, and intermediate commercials [30, 46, 51]. Due to the long duration, repeated contents, and unexpected interruption, they always affect user experiences [37, 51]. A public vote [13] shows that 95% users wish video content providers (VCPs) to assist them in skipping redundant clips, since manually skipping is frustrating—they always jump over the contents too short or too far [1, 26], on account of varied locations of redundant clips [30, 59].

Driven by users' demands [8, 69], many VCPs like Netflix and Disney+ have provided functionalities for skipping redundant clips. They either automatically skip them, or display a "skip" button. However, these seemingly user-friendly functions are far from satisfactory in practice, which have received plenty of negative comments [2, 9] from users. Some inexperienced users even doubt that these functions lead to failure in the whole streaming process [12].

**Measurement study.** To quantitatively understand why the seemingly easy-to-mark redundant clips incur undesirable user experiences, we conduct the first large-scale, crowdsourced measurement study on redundancy skipping functions of eight popular VCPs. These VCPs include Netflix, Amazon Prime Video, Disney+, Hulu, HBO Max, iQIYI, Tencent Video, and Youku, covering more than 90% of the global market share in 2022 [14, 47, 48]. Note that all the studied VCPs have provided the functionality for users to skip redundant clips. For each VCP, we randomly sample 30% sets of correlated videos from different genres in their libraries available in the USA as of Oct. 2022. In total, we have 145,976 correlated videos in our dataset. Table 2 lists the detailed statistics of them.

We next collect the timestamps of redundant clips marked by the VCPs. An intuitive method is to monitor the DOM tree [45, 70] of the playback web page, so as to examine when the "skip" button appears and which time point the player seeks to after the skip action is triggered. However, this approach is too time-consuming since it needs to play the entire video (each costs tens of minutes). To address this, we reverse engineer the streaming process of these VCPs, and observe that all VCPs deliver their marked timestamp information to the client browser during the playback page loading (for initializing the player). Thus, we collect marker information by deploying a puppet web browser to intercept network requests and responses during playback page loading using the Puppeteer library [11]. This enables us to gather marker information in merely 6 seconds on average for a video (*i.e.,* ~10 days in total).

We then check the correctness of the collected markers. Given the subjectivity in recognizing redundant clips, we conduct a crowdsourcing study by recruiting 25 volunteer users of different genders, ages (ranging from 21 to 48), and education levels (from undergraduate to PhD) to help examine VCPs' markers. Specifically, we distribute the videos in our dataset to them, and ask them to watch the videos over the web to check whether redundant clips are skipped correctly. If not, they are asked to manually mark the timestamp. To improve efficiency, we sample video frames every five seconds as a thumbnail hint for them, based on which they can quickly pinpoint redundancy locations. Each video is examined by two volunteers, and we average the timestamps if they both mark the clips. If their judgments are inconsistent, we will participate to resolve it. We also develop a browser plug-in that extracts the `currentTime` property [17] from the `video` tag of playback pages, so that the volunteers can mark redundant clips on a millisecond basis.

*This study was conducted under a well-established IRB, and does not raise any ethical issues.*

**Measurement findings.** By analyzing the crowdsourcing results, we have multi-fold findings on the prevalence and specific patterns (listed in Table 1) of the unexpected experience of redundant clips in correlated web video streaming.

First, we find that marking visual redundancy is in fact an error-prone job for VCPs. Figure 2 shows the percentage of video episodes

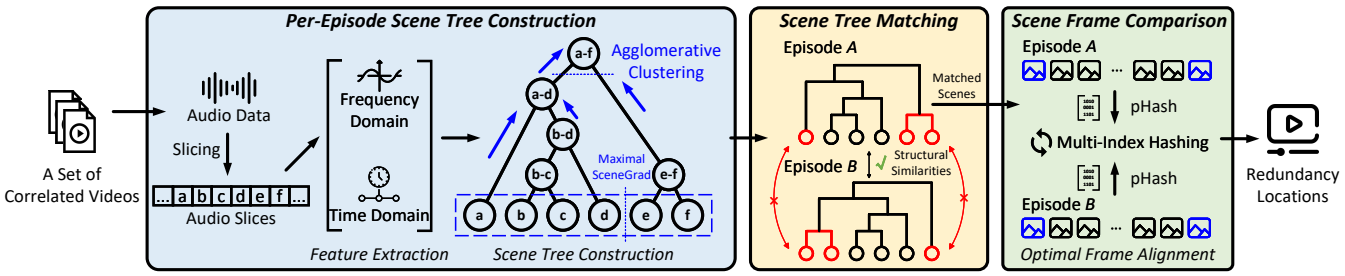


**Figure 5: Architectural overview of SkipStreaming.**

that have our identified symptoms. Users can experience redundancy skipping issues in all the studied VCPs. Among them, Youku induces the worst experience, with 54.7% of videos (that have redundant clips) marked improperly. Besides, iQIYI, Tencent Video, and Youku have a high circumscribed skip rate (>20%) since they only mark redundancy in heading and tailing parts of videos. Also, as the most popular VCP in recent years, Netflix incurs the least issues, but there are still 16.7% episodes having problematic markers.

Further, we find the main experience issue to be that VCPs often mark redundant clips probabilistically or inaccurately. As shown in Figure 3, redundancy in a considerable portion (35.8% on average for all VCPs) of video series is skipped probabilistically. This is especially the case for Hulu, where over half of the series have at least 8% episodes whose redundant clips cannot be skipped. We attribute this to the fact that the "skip function" in Hulu has only been available for two years [65], and many redundant clips still have not been marked. Figure 4 shows the marker deviation of each VCP. Except for Tencent Video that has an acceptable (81%) accuracy, the other VCPs' markers deviate from the right time for 5.72 seconds on average. Among these problematic markers, those outside the redundant clips will cause users to miss video plots. Note that even a single plot miss can significantly disturb users.

**VCPs' dilemma and design challenges.** To demystify the root causes of these symptoms, we extensively survey the literature, and report the symptoms to relevant VCPs. Fortunately, iQIYI gets back to our report and offers us an opportunity to interview with their development team. We obtain a number of findings on the methods adopted by VCPs and the real-world dilemma they encounter.

Traditional byte-level data redundancy detection techniques [68] lose efficacy for video clips, as different episodes have their unique encoding parameters [27, 33]. Also, although CV methods that perform frame-wise analysis are more accurate, they consume tremendous computation resources. This is unbearable for top VCPs that publish tens of thousands of high-resolution videos every day (*e.g.,* Youku and iQIYI). Besides, model pruning techniques that reduce the number of parameters for CV models sacrifice the detection coverage and accuracy, leading to Pattern 1 and 3 listed in Table 1. Further, as audio fingerprinting techniques are vulnerable to fingerprint collisions and information loss, it is difficult to determine a generic similarity threshold for all videos. This makes the accuracy unpredictable, causing Pattern 1 and 2. The intros of some episodes may even be recognized as outros when collisions occur, leading to functional failure in the web player and causing Pattern 4.

# 3 DESIGN AND IMPLEMENTATION

## 3.1 System Overview

We present SkipStreaming, an accurate and lightweight visual redundancy detection system for correlated videos. Our main idea is to rethink about how the contents of correlated videos are organized during video production. Specifically, today's correlated videos are pervasively organized based on *scene* [40, 66]. A scene is part of a video where a sequence of logically relevant events occur, which is a widely-used concept in film making for producing a coherent and intriguing storyline.

The notion of scenes provides us with a unique opportunity to detect visual redundancy—the occurrence of intros, outros, recaps, and commercials (no matter where and how long they are) is always accompanied by scene changes. This is quite understandable since they are inserted into the main story as content-wise independent segments during video production. As a result, redundant clips will not begin or end arbitrarily within a certain scene.

With the above scene insight, we build SkipStreaming based *audio-guided scene sketch*, a multimodal (*i.e.,* acoustic-narrative-visual) fusion methodology. The key idea is that different scenes usually have different characters' speaking, ambient noises, background music, *etc.* SkipStreaming roughly sketches scenes by tracking acoustic variation in videos. Thereby, it can efficiently compare a small number of video frames near scene transitions to locate redundancy. Figure 5 depicts the workflow of SkipStreaming, which takes a set of correlated videos as the input, and performs the following steps to recognize redundant clips:

- *Per-Episode Scene Tree Construction* (§3.2). To quickly identify scene transitions using audio data, we divide the audio into slices and devise a novel acoustic feature named *scene-aware acoustic gradient* (denoted as SceneGrad) to differentiate the acoustic characteristics of the slices over time. Given that a redundant clip may contain multiple similar coherent scenes, we further recursively cluster scenes together by minimizing their holistic SceneGrad. Such clustering process outputs a tree structure (*i.e.,* scene tree) for each episode, which models the inclusion relationship between its probable redundant clips and scenes.
- *Inter-Episode Scene Tree Matching* (§3.3). After the scene tree of each episode is constructed, SkipStreaming performs scene tree matching between each pair of episodes to find potential redundant clips that have similar scene organizations. To this end, it examines the structural similarity of their common subtrees. As conventional tree matching approaches bear a high time

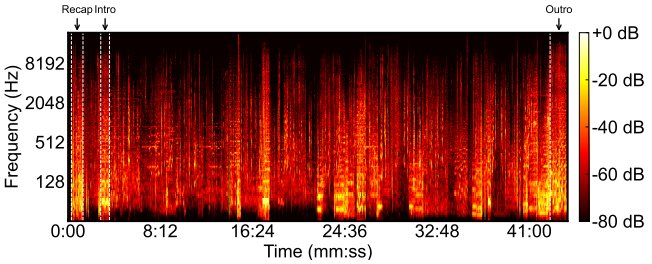


**Figure 6: Power spectrum of the frequency components of a TV episode's audio data after Fourier transform.**

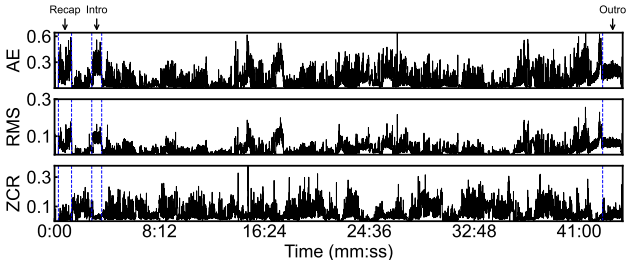


**Figure 7: Time-domain features of a TV episode's audio data.**

complexity, SkipStreaming takes advantage of the temporal and structural relationships among tree nodes for an approximate matching result. Consequently, it picks out candidate redundant scene sequences in different episodes, with the same duration and similar scene organizations.

- *Inter-Episode Scene Frame Comparison* (§3.4). In order to determine whether the above candidate scene sequences are truly redundant, SkipStreaming performs fine-grained video frame matching near their scene transitions. It decompresses video frames that embrace the scene transitions on demand, and finds the optimal frame alignment based on the hamming distance of frames' pHash [71]. We accelerate this process using the *multi-index hashing* [53] technique, which caches intermediate comparison results to avoid repetitive frame comparisons.

## 3.2 Per-Episode Scene Tree Construction

**Acoustic feature extraction.** Identifying the scene organization in an episode is a crucial preliminary step for efficient redundancy recognition. As mentioned before, different scenes usually present different acoustic characteristics given their unique audio content composition. Specifically, Figure 6 shows the decomposed frequency components of the audio of an episode of the TV series *The Coyotes* [20]. The frequency-domain components of the recap, intro, and outro differ significantly from those of the neighboring scenes. Besides, Figure 7 shows the episode's time-domain audio features, *i.e.,* amplitude envelope (AE), root mean square energy (RMS), and zero-crossing rate (ZCR). There is also a clear boundary between three redundant clips and their neighboring scenes. For example, the AEs of the recap and the intro are around 0.2 and 0.3 respectively, while that of the main story in between is around 0.08. Our extensive investigation involving 200 randomly-sampled correlated videos indicates that such dissimilarity in scenes' acoustic features widely exists. On average, the variance of the acoustic features at a scene transition is 5.4 times higher than those inside a scene.

We thus extract both frequency-domain and time-domain audio features for lightweight scene representation and to prepare for scene transition detection. For each episode, we split its audio data into equal-length and half-overlapped slices. We set the slice gap to be the duration of a video frame, so that each slice is aligned to a frame at the same time point. For each slice, we use the first 13 of mel-frequency cepstral coefficients (MFCCs) as the frequency-domain feature, and AE, RMS, and ZCR as time-domain features.

**Scene tree construction based on SceneGrad.** After audio slices' features are extracted, we use them to infer the scene organization of the episode. As a redundant clip can contain multiple coherent scenes, we wish to model the coherence of scene sequences, so as to quickly pinpoint redundant clips in the next steps (§3.3 and §3.4). To this end, we perform agglomerative clustering [50] to merge temporally-adjacent and acoustically-similar audio slices into macro-level scene clusters (each contains one or more scenes). During the merging process, we minimize the feature variance within the same cluster, while maximizing the variance between different neighboring clusters.

We devise a novel acoustic feature called *scene-aware acoustic gradient* (denoted as SceneGrad) to quantify such scene cluster variance, which acts as the linkage function [22] for agglomerative clustering. Instead of simply differentiating the characteristics of two adjacent audio slices, SceneGrad measures the weighted variance in audio slice features within a scalable time window. This enables us to capture both local and holistic audio feature variation, so as to largely avoid false predictions.

Formally, given two adjacent clustered audio slice sequences $P = \{p_1, p_2, \cdots, p_m\}$ and $Q = \{q_1, q_2, \cdots, q_n\}$, with their feature vectors to be $V_P = \{v_{p_1}, v_{p_2}, \cdots, v_{p_m}\}$ and $V_Q = \{v_{q_1}, v_{q_2}, \cdots, v_{q_n}\}$, SceneGrad between this two clusters is defined as

$$SceneGrad\,(P, Q) = \frac{1}{|P \cup Q|}\left(\sum_{k \in P \cup Q} w_k \cdot \|v_k - \mu_{PQ}\|^2\right), \quad (1)$$

where $\mu_{PQ}$ is the centroid of all feature vectors in $V_P$ and $V_Q$. $w_k$ is the weight value for audio slice $k$, which increases as the location of audio slice approaches the boundary between the two clusters $P$ and $Q$. Specifically, suppose that $P$ is ahead of $Q$ in the original video episode. For audio slice $k \in P \cup Q$, we assign the weight value

$$w_k = \begin{cases} \frac{i}{1+2+\cdots+m}, k = p_i \in P, \\ \frac{n+1-j}{1+2+\cdots+n}, k = q_j \in Q. \end{cases} \quad (2)$$

Compared with traditional similarity measures used in agglomerative clustering such as complete linkage and single linkage [50], SceneGrad has a special focus on the smoothness of audio characteristics within the same scene. This is achieved by assigning a larger weight value to the adjacent audio slices that locate between two clusters during the merging process. In this way, the similarity or the difference between the connecting audio slices of the two clusters will be magnified. As a result, slices with smooth changes in characteristics have a higher priority to be clustered together.

In each iteration of agglomerative clustering, SkipStreaming selects and merges two adjacent clusters with the minimal SceneGrad increase after merging. If the SceneGrad exceeds a threshold $T$ after merging, we mark the original two clusters as two complete scenes.

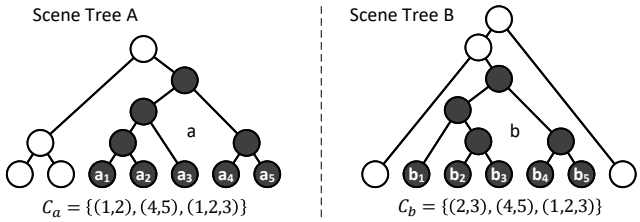


**Figure 8: Matching coherent scene sequences across scene trees.**

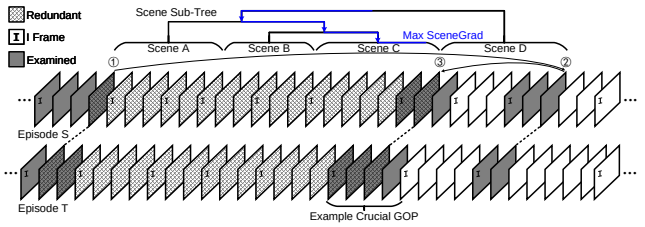


**Figure 9: Frame matching and alignment between matched scenes.**

In practice, $T$ is set to be 0.56 times of the variance of all audio slice features, which achieves a fine F1 score of 93% in identifying scenes. As SkipStreaming continues to merge audio slices, more clusters will be marked as complete scenes, and the adjacent coherent scenes will be further clustered as a *coherent scene sequence*. Such sequence is likely to be a complete redundant clip due to its smooth change in acoustic characteristics.

The above scene sequence clustering process finally outputs a hierarchical structure of the input video episode (*i.e.,* the scene tree as shown in Figure 5). Each node in the scene tree represents a time interval in the original episode. The derivation of two child nodes from a parent node indicates the transition of the two adjacent scenes. If the parent video segment has redundant clips, the scene transition point between the two child nodes is the most likely to be the beginning or ending of the redundancy, given its most significant acoustic feature changes manifested by SceneGrad.

## 3.3 Inter-Episode Scene Tree Matching

With the scene tree of each episode, SkipStreaming performs scene tree matching among all episodes to pick out probable redundant clips. This step is also important because if we can rule out most non-redundant scene sequences, the subsequent frame matching will avoid considerable unnecessary frame comparisons. Unfortunately, traditional tree matching algorithms based on edit distance such as RTED [54] and APTED [55] are not much efficient. It has been proved that the tree edit distance problem cannot be solved in a complexity lower than $O(n^3)$ [25], where $n$ is the number of tree nodes. This makes them infeasible for the pairwise comparison among all episodes with tens of thousands of scene tree nodes.

We address the problem by using temporal relationships among tree nodes as a heuristic to reduce the complexity. Specifically, if two clips are visually redundant, they must have the same duration and scene organization. For a pair of episodes to be compared, we first extract sub-trees from their scene trees that have the same number of leaf nodes. As illustrated in Figure 8, nodes in shadow are the example extracted sub-trees $a$ and $b$ from scene trees $A$ and $B$, respectively. Note that there may be multiple disjoint sub-trees with the same number of leaf nodes between each two episodes.

The above procedure benefits the matching process in two aspects. First, it ensures that the candidate redundant scene sequences from two video episodes have the same time duration, since each leaf node represents an equal-length audio slice (mentioned in §3.2). Second, by only considering sub-trees, we can significantly reduce the number of tree nodes to be compared. The time complexity of sub-tree extraction is only $O(m+n)$ the using hash map [32], where $m$ and $n$ are the number of all tree nodes in two scene trees.

Due to lossy audio encoding, the constructed scene tree of redundant clips may slightly differ. Thus, we next match the sub-trees to measure the similarity of two episodes' scene organization. We first sequentially number the leaf nodes of sub-trees ($a$ and $b$ in Figure 8), and then make use of the intermediate results of the clustering process by denoting $C$ as all leaf node clusters. For example in Figure 8, $C_a$ is $\{(1,2),(4,5),(1,2,3)\}$, and $C_b$ is $\{(2,3),(4,5),(1,2,3)\}$. We calculate the structural similarity between sub-tree $a$ and $b$ as

$$StructSIM(a,b) = \frac{2 \cdot |C_a \cap C_b|}{|C_a| + |C_b|}. \quad (3)$$

In practice, if StructSIM exceeds a tuned threshold of 0.72, we determine that the video segments have similar scene organization. In this way, we rule out the non-redundant clips that are different in scene organization or time duration.

## 3.4 Inter-Episode Scene Frame Comparison

With the matched scene trees and the corresponding video segments of all episodes of the series, SkipStreaming performs fine-grained video frame matching on them to check if they are (or if they contain) visual redundancy and outputs the marked timestamps.

**Detecting redundant frames with pHash.** As different videos are encoded with different parameters, even a pair of frames from two episodes are redundant, they may have pixel-level differences. Thus, to determine whether frames are matched, we extract pHash from the frames and calculate their hamming distance. Unlike common pHash algorithms that omit the high-frequency components of discrete cosine transform (DCT) coefficients for accommodating manipulation in images [71], we preserve all coefficients. This enables SkipStreaming to be sensitive to the pixel changes.

We now detail how we match video frames. As shown in Figure 9, suppose that we have two matched video segments (between the two outermost dashed lines) of Episode $S$ and $T$ with the same scene sub-tree structure[1]. Starting from the root node of the sub-tree, SkipStreaming compares their video frames near scene transitions (recall that an interior scene tree node represents a scene transition). In the example, we first examine frames near the beginning of scene A and ending of scene D, and find that frame ① is redundant while frame ② is not. Thus, we search the ending position of the redundant clip beginning at ①, by examining the frame near the maximal SceneGrad (③). In this way of recursive frame comparisons over the scene tree, we finally determine the exact timestamps.

In an encoded video file, except for I frames, the decoding of P and B frames must depend on its precedent decoded frames [58].

[1]For simplicity, we let the example scene sequences have the same scene sub-tree. For those whose sub-trees are not the same, we perform frame matching on each sub-tree.

Therefore, when comparing frames near scene transitions, we examine all frames between two adjacent I frames that embrace the scene transition, so as to accurately align redundant frames. We call these frames a crucial group of pictures (or crucial GOP for short). SkipStreaming aligns frame sequences in two crucial GOPs by minimizing the average pHash hamming distance of the frames.

**Faster frame comparisons using multi-index hashing.** A redundant clip may always appears in different episodes. To avoid repetitive comparisons for the same redundant clip across episodes, we cache pHash of examined frames using multi-index hashing (MIH) [53]. When SkipStreaming compares video frames, it adds their pHash into multi-index hash table. If the clip is redundant, it will record the frame's time offset in the clip. For an episode whose frames have not been compared, SkipStreaming first looks up its pHash in the hash table. Once a frame found in the table is marked as redundant, SkipStreaming directly marks the clip as redundant and generates the timestamps based on the recorded time offset.

# 4 EVALUATION

## 4.1 Experiment Setup

We evaluate SkipStreaming in terms of accuracy and time overhead by comparing it with three state-of-the-art video redundancy detection systems, namely Jellyfin Skipper [16], Kodi Detector [18], and Amazon B-LSTM [37]. Among them, Jellyfin Skipper is an audio fingerprinting-based approach, while Kodi Detector is a pure CV-based one. In contrast, Amazon B-LSTM is a hybrid of CV and audio fingerprinting techniques based on Bi-LSTM [57].

Our evaluation involves 16,238 episodes from 200 video series collected on mainstream VCPs (9,251 hours in total). They cover a wide variety of genres such as anime, dramas, and documentaries. We run each solution on top of a commodity server with an Intel Xeon 10-core CPU@2.4 GHz, 64-GB DDR memory, 960-GB SSD, and no GPU. We use the same methodology described in §2 to assess the quality of the generated markers from users' perspectives. If the recruited users believe that the timestamp is correct, it is regarded as a true positive. If they think that the timestamp deviates from the right position, it is a false positive. If they find a clip is redundant but not marked, then it is a false negative.

## 4.2 Detection Performance

**Detection accuracy.** Table 3 shows the overall detection accuracy of the four approaches. Jellyfin Skipper yields the worst accuracy, with the precision and recall being <70%. This is because the audio fingerprinting technique is vulnerable to the noise introduced by audio encoding, which makes the performance unstable across different video series. Even if we fine-tune the fingerprint similarity threshold to tolerate the noise for a higher recall (70%), the incurred fingerprint collisions will significantly degrade the precision (61%).

For the two CV-based approaches, Kodi Detector and Amazon B-LSTM, they have slightly better detection performance than Jellyfin Skipper, but are still far from satisfactory (F1 score <0.8). Delving deeper, we find that they both sacrifice the detection accuracy for efficiency in their design. Kodi Detector samples video frames that differ significantly from their precedent frames. It then regards all frames between two sampled frames as redundant if they match the frames in another video. This incurs a number of false positives if the sampled frames are black frames between shot changes. Also, Amazon B-LSTM simply samples one visual and audio feature per second for model inference, without any inter-episode comparisons.

**Table 3: Accuracy of the four solutions. # TP denotes the number of true positives. # RC denotes the detected number of redundant clips.**

| Solution | Precision | Recall | F1 Score | # TP | # RC |
|---|---|---|---|---|---|
| Jellyfin | 68% | 62% | 0.65 | 17,848 | 26,069 |
| Kodi | 76% | 73% | 0.75 | 21,280 | 27,868 |
| Amazon | 81% | 75% | 0.78 | 21,691 | 26,864 |
| SkipStreaming | **98%** | **93%** | **0.96** | **27,083** | 27,533 |

In contrast, SkipStreaming has the best accuracy (precision and recall >90%) owing to its meticulous use of lightweight audio data to sketch scenes and guide video comparisons. Of course, it incurs false predictions in practice. Figure 10 shows the time deviation of the four solutions when false positives occur. SkipStreaming incurs the minimal time deviation (770 ms on average). By examining these false positives, we find that they are mainly caused by *gradual visual transitions* (*e.g.,* fades and dissolves) between scenes. This makes scene boundaries rather vague, and thus the detected time points can slightly deviate from the user-perceived ones. Fortunately, the contents of these false positives are not important to the story, so they hardly affect user experiences. For false negatives, they are too short (<5 seconds) and have no significant acoustic characteristic changes. In practice, they hardly affect user experiences either.

**System overhead.** Figure 11 demonstrates the time overhead of the four solutions. Jellyfin Skipper presents the lowest time overhead (but the worst accuracy as discussed above) with an average of 26.7 seconds. This is because it only uses lightweight audio fingerprinting techniques. In comparison, the average time overhead of SkipStreaming is 38.2 seconds, which is a bit higher than that of Jellyfin Skipper due to its video frame processing. Nevertheless, considering its significant accuracy improvement, we believe that such time consumption increase is worthwhile.

The two CV-based approaches incur prohibitively high time overhead. Kodi Detector takes 371.5 seconds on average to process a single episode, while Amazon B-LSTM takes 524.7 seconds. This is not surprising because they both go through the entire video and perform heavyweight visual feature extraction and analysis. On the other side, although SkipStreaming also involves video frame processing, it selectively examines crucial video frames under the guidance of audio data, thus achieving acceptable time overhead.

**Performance on public datasets.** To further demonstrate SkipStreaming's efficacy, we evaluate it on the public BBC Planet Earth dataset [24], which includes 11 episodes from the BBC's documentary and has been used by many studies in scene segmentation and understanding for performance evaluation. SkipStreaming can detect all redundant clips in the dataset with an average time deviation of 542 ms and an average time cost of 41 seconds.

## 4.3 Ablation Study

We perform an ablation study to understand the contributions of SkipStreaming's each component to the overall efficacy.

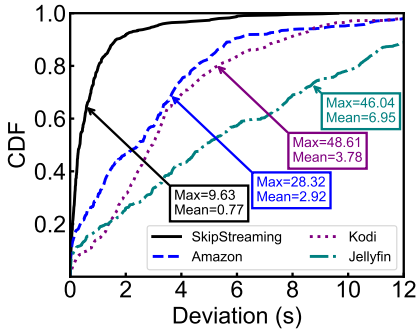


**Figure 10: Time deviation of four solutions.**

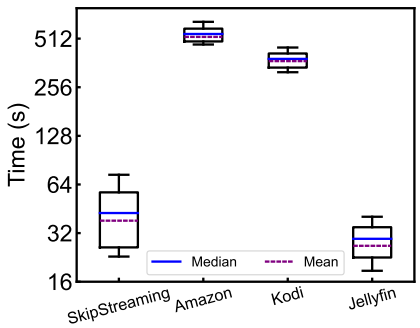


**Figure 11: Time overhead of four solutions.**

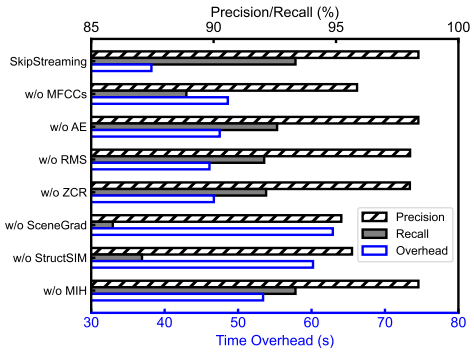


**Figure 12: Ablation study for SkipStreaming.**

**Frequency-domain and time-domain features.** As shown in Figure 12, when we remove MFCCs, the precision and recall decrease to 96% and 89% respectively, and the overhead increases by 27%. Also, when we remove any time-domain feature, the accuracy degrades by ~1%, and the overhead increases by over 20%. This indicates that the extracted audio features are all necessary for the accuracy and efficiency. In fact, removing audio features results in inaccurate scene transition detection. When false negatives of scene transition detection occur, the redundant clips near them will not be detected, leading to probabilistic skips. For false positives of scene transition detection, they will cause subsequent steps to process more video frames, and finally increase the time overhead.

**SceneGrad and StructSIM.** Disabling SceneGrad and StructSIM significantly degrades the performance. When SceneGrad is replaced with the complete linkage function, the precision and the recall drop by 3% and 7% respectively, with a 65% increase in overhead. This indicates that SceneGrad well depicts the scene of videos. Without it, scenes are segmented incorrectly, which adds unnecessary visual data analysis to frame matching. Besides, disabling StructSIM causes a 2% drop in precision and a 6% drop in recall, with a 58% increase in overhead. Thus, StructSIM is important for scene tree matching to combat the introduced noise of audio encoding.

**Multi-index hashing.** As shown in Figure 12, removing MIH will not affect the detection accuracy of SkipStreaming, but significantly (40%) increases the overhead. This is because there is considerable repeated video frame decoding and matching during inter-episode comparisons, which incurs a lot of additional computation. By caching and indexing these video frames using MIH, we can further achieve significant performance improvement.

## 5 RELATED WORK

**Visual redundancy detection.** There have been quite a few methods for automatically detecting visual redundancy among videos. ViDeDup [39] is an application-aware video redundancy detection system, which recognizes visually similar video sequences among web videos based on video signature that depicts video's color and spatial-temporal distributions. Huang *et al.* [38] transform a video into a one-dimensional video distance trajectory, and detect visual redundancy by comparing a sequence of compact signatures. Other studies [42, 44, 67] propose whole-video redundancy detection schemes to reduce duplicate results for web video retrieving.

Different from these works, SkipStreaming detects visual redundancy from correlated videos, and focuses much on the timing accuracy of the generated redundancy markers.

**Video scene segmentation and understanding.** Segmenting and understanding scenes from videos has long been a challenging task. Towards this goal, many CV-based approaches have been proposed [29, 56, 64], which typically cluster coherent frame/shot sequences using specially extracted visual features, and then feed the features into Bi-LSTM for temporal analysis. Also, some VCPs [15, 21] record users' seek actions on the player progress bar during playback to roughly segment scenes. Further, to facilitate the research on video scenes, a number of datasets have been built and publicly released. For example, the MovieScenes [56] dataset contains 21K annotated scene segments from 150 movies, while BBC Planet Earth [24] derives from the episodes of BBC documentaries.

In comparison, SkipStreaming makes use of lightweight audio data of videos to efficiently sketch scenes and identify scene transitions. Such scene transition information is then strategically used as guidance to speed up the detection of redundant video frames.

## 6 CONCLUSION

This paper presents SkipStreaming, an accurate and lightweight user-oriented redundancy detection system for correlated videos streaming over the web. The seemingly easy-to-mark redundant clips have long been existent and harming user experiences, while existing detection approaches are either inaccurate or inefficient. We address the problem from the novel perspective of *scenes*, which are the basic story units that compose a video. SkipStreaming extracts scene information via our specially-designed *audio-guided scene sketch* methodology, and selectively compares a small portion of video frames to quickly detect visual redundancy. Our evaluations on numerous real-world correlated videos confirm its efficacy.

## ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable comments and suggestions. Also, we thank Baolin Tian for his help in the measurement study, as well as Zhe Tang for proofreading the paper. This work is supported in part by National Key R&D Program of China under grant 2022YFB4500703, National Natural Science Foundation of China under grants 61902211 and 62202266, China Postdoctoral Science Foundation under grant 2022M721831, and Microsoft Research Asia under grant 100336949.

# REFERENCES


[1] 2018. UX in Daily Life - Netflix's "Skip Intro". https://www.linkedin.com/pulse/ux-daily-life-netflixs-skip-intro-imran-razaq/.

[2] 2019. The Intros of Some Episodes in Disney+ Cannot Be Skipped. https://www.reddit.com/r/DisneyPlus/comments/dv6369/disney_has_a_skip_intro_function/f7au5dh/.

[3] 2020. Cisco Annual Internet Report - Cisco Annual Internet Report (2018–2023) White Paper. https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.

[4] 2020. Go Ahead and Skip That Intro. https://www.plex.tv/blog/go-ahead-and-skip-that-intro/.

[5] 2021. Accessibility Features on Disney+. https://help.disneyplus.com/csp?id=csp_article_content&sys_kb_id=bc24b6abdb054090fbf26ac2ca961950.

[6] 2021. Disney+. https://www.disneyplus.com.

[7] 2021. Google Video AI. https://cloud.google.com/video-intelligence.

[8] 2021. Here's How Much Time You Can Save by Skipping TV Intros. https://www.lifehacker.com.au/2021/06/save-time-skipping-tv-intros/.

[9] 2021. iQIYI and Tencent Video Responded to the Problem That Subscribers Still Have to Watch Ads: You Can Drag the Progress Bar to Skip. https://lujuba.cc/en/614223.html.

[10] 2021. Netflix. https://www.netflix.com/.

[11] 2021. Puppeteer: Headless Chrome Node.js API. https://github.com/puppeteer/puppeteer.

[12] 2021. T.O.T.S. Season 2 Is Steaming Incorrectly. https://www.reddit.com/r/DisneyPlus/comments/rkkzqw/tots_season_2_is_steaming_incorrectly_we_are/.

[13] 2022. Auto "Skip Intro" Option? https://www.reddit.com/r/PleX/comments/rvunb5/auto_skip_intro_option/.

[14] 2022. China 2022 Survey Report: iQIYI, Tencent Video Tops in Online Video. https://www.spglobal.com/marketintelligence/en/news-insights/research/china-2022-survey-report-iqiyi-tencent-video-tops-in-online-video.

[15] 2023. Bilibili. https://www.bilibili.com/.

[16] 2023. The Free Software Media System | Jellyfin. https://jellyfin.org/.

[17] 2023. HTMLMediaElement: currentTime Property - Web APIs | MDN. https://developer.mozilla.org/en-US/docs/Web/API/HTMLMediaElement/currentTime.

[18] 2023. Kodi Open Source Home Theater Software. https://kodi.tv/.

[19] 2023. Watch Naruto | Netflix. https://www.netflix.com/title/70205012.

[20] 2023. Watch The Coyotes | Netflix. https://www.netflix.com/title/81493687.

[21] 2023. YouTube. https://www.youtube.com/.

[22] Margareta Ackerman and Shai Ben-David. 2016. A Characterization of Linkage-Based Hierarchical Clustering. *The Journal of Machine Learning Research* 17, 1 (2016), 8182–8198.

[23] Adam Hayes. 2023. YouTube Stats: Everything You Need to Know in 2023. https://www.wyzowl.com/youtube-stats/.

[24] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. 2015. A Deep Siamese Network for Scene Detection in Broadcast Videos. In *Proc. of ACM MM*. 1199–1202.

[25] Karl Bringmann, Paweł Gawrychowski, Shay Mozes, and Oren Weimann. 2020. Tree Edit Distance Cannot Be Computed in Strongly Subcubic Time (Unless APSP Can). *ACM Transactions on Algorithms* 16, 4 (2020), 1–22.

[26] Cameron Johnson. 2022. Looking Back on the Origin of Skip Intro Five Years Later. https://about.netflix.com/en/news/looking-back-on-the-origin-of-skip-intro-five-years-later.

[27] Chao Chen, Yao-Chung Lin, Steve Benting, and Anil Kokaram. 2018. Optimized Transcoding for Large Scale Adaptive Streaming Using Playback Statistics. In *Proc. of IEEE ICIP*. 3269–3273.

[28] Liang-Hua Chen, Yu-Chun Lai, and Hong-Yuan Mark Liao. 2008. Movie Scene Segmentation Using Background Information. *Pattern Recognition* 41, 3 (2008), 1056–1065.

[29] Shixing Chen, Xiaohan Nie, David Fan, Dongqing Zhang, Vimal Bhat, and Raffay Hamid. 2021. Shot Contrastive Self-Supervised Learning for Scene Boundary Detection. In *Proc. of IEEE/CVF CVPR*. 9796–9805.

[30] Zhi-Qi Cheng, Yang Liu, Xiao Wu, and Xian-Sheng Hua. 2016. Video Ecommerce: Towards Online Video Advertising. In *Proc. of ACM MM*. 1365–1374.

[31] Sam Cook. 2022. 50+ Netflix Statistics, Facts and Figures in 2023. https://www.comparitech.com/blog/vpn-privacy/netflix-statistics-facts-figures/.

[32] William H. E. Day. 1985. Optimal Algorithms for Comparing Trees with Labeled Leaves. *Journal of Classification* 2, 1 (1985), 7–28.

[33] Jan De Cock, Zhi Li, Megha Manohara, and Anne Aaron. 2016. Complexity-Based Consistent-Quality Encoding in the Cloud. In *Proc. of IEEE ICIP*. 1484–1488.

[34] Shaojin Ding, Tianlong Chen, and Zhangyang Wang. 2022. Audio Lottery: Speech Recognition Made Ultra-Lightweight, Noise-Robust, and Transferable. In *Proc. of ICLR*.

[35] Boyuan Feng, Yuke Wang, Gushu Li, Yuan Xie, and Yufei Ding. 2021. Palleon: A Runtime System for Efficient Video Processing toward Dynamic Class Skew. In *Proc. of USENIX ATC*. 427–441.

[36] A. Hampapur, T. Weymouth, and R. Jain. 1994. Digital Video Segmentation. In *Proc. of ACM MM*. 357–364.

[37] Xiang Hao, Kripa Chettiar, Ben Cheung, Vernon Germano, and Raffay Hamid. 2021. Intro and Recap Detection for Movies and TV Series. In *Proc. of IEEE WACV*. 167–176.

[38] Zi Huang, Heng Tao Shen, Jie Shao, Bin Cui, and Xiaofang Zhou. 2010. Practical Online Near-Duplicate Subsequence Detection for Continuous Video Streams. *IEEE Transactions on Multimedia* 12, 5 (2010), 386–398.

[39] Atul Katiyar and Jon Weissman. 2011. ViDeDup: An Application-Aware Framework for Video De-Duplication. In *Proc. of USENIX HotStorage*.

[40] Ephraim Katz and Ronald Dean Nolen. 2012. *The Film Encyclopedia 7th Edition: The Complete Guide to Film and the Film Industry*. Collins Reference.

[41] J.R. Kender and Boon-Lock Yeo. 1998. Video Scene Segmentation via Continuous Video Coherence. In *Proc. of IEEE CVPR*. 367–373.

[42] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Ioannis Kompatsiaris. 2019. ViSiL: Fine-Grained Spatio-Temporal Video Similarity Learning. In *Proc. of IEEE/CVF ICCV*. 6351–6360.

[43] Zhenhua Li, Yafei Dai, Guihai Chen, and Yunhao Liu. 2023. *Content Distribution for Mobile Internet: A Cloud-Based Approach, Second Edition*. Springer Nature Press.

[44] Jiajun Liu, Zi Huang, Hongyun Cai, Heng Tao Shen, Chong Wah Ngo, and Wei Wang. 2013. Near-Duplicate Video Retrieval: Current Research and Future Trends. *Comput. Surveys* 45, 4 (2013), 44:1–44:23.

[45] Wei Liu, Xinlei Yang, Hao Lin, Zhenhua Li, and Feng Qian. 2022. Fusing Speed Index during Web Page Loading. In *Proc. of ACM SIGMETRICS*. 1–23.

[46] Yao Liu, Sam Blasiak, Weijun Xiao, Zhenhua Li, and Songqing Chen. 2015. A Quantitative Study of Video Duplicate Levels in YouTube. In *Proc. of Springer PAM*. 235–248.

[47] Amanda D. Lotz. 2022. *Netflix and Streaming Video: The Business of Subscriber-Funded Video on Demand*. John Wiley & Sons.

[48] Amanda D Lotz, Oliver Eklund, and Stuart Soroka. 2022. Netflix, Library Analysis, and Globalization: Rethinking Mass Media Flows. *Journal of Communication* 72, 4 (2022), 511–521.

[49] Ben Munson. 2019. Nearly 75% of U.S. Households Have Either Netflix, Hulu or Amazon Prime Video. https://www.fiercevideo.com/video/nearly-75-u-s-households-have-either-netflix-hulu-or-amazon-prime-video.

[50] Fionn Murtagh and Pedro Contreras. 2012. Algorithms for Hierarchical Clustering: An Overview. *WIREs Data Mining and Knowledge Discovery* 2, 1 (2012), 86–97.

[51] Maryam Nematollahi and Xiao-Ping Zhang. 2014. Automatic Video Intro and Outro Detection on Internet Television. In *Proc. of ACM WISMM*. 43–46.

[52] Netflix. 2021. How to Skip Intros on TV Shows. https://help.netflix.com/en/node/63402.

[53] Mohammad Norouzi, Ali Punjani, and David J. Fleet. 2014. Fast Exact Search in Hamming Space with Multi-Index Hashing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 6 (2014), 1107–1119.

[54] Mateusz Pawlik and Nikolaus Augsten. 2011. RTED: A Robust Algorithm for the Tree Edit Distance. *Proceedings of the VLDB Endowment* 5, 4 (2011), 334–345.

[55] Mateusz Pawlik and Nikolaus Augsten. 2015. Efficient Computation of the Tree Edit Distance. *ACM Transactions on Database Systems* 40, 1 (2015), 1–40.

[56] Anyi Rao, Linning Xu, Yu Xiong, Guodong Xu, Qingqiu Huang, Bolei Zhou, and Dahua Lin. 2020. A Local-to-Global Approach to Multi-Modal Movie Scene Segmentation. In *Proc. of IEEE/CVF CVPR*. 10143–10152.

[57] Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional Recurrent Neural Networks. *IEEE Transactions on Signal Processing* 45, 11 (1997), 2673–2681.

[58] Susanna Schwarzmann, Nick Hainke, Thomas Zinner, Christian Sieber, Werner Robitza, and Alexander Raake. 2020. Comparing Fixed and Variable Segment Durations for Adaptive Video Streaming: A Holistic Analysis. In *Proc. of ACM MMSys*. 38–53.

[59] Roger Silverstone. 1984. Narrative Strategies in Television Science—A Case Study. *Media, Culture & Society* 6, 4 (1984), 377–410.

[60] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *Proc. of ICLR*.

[61] Kevin Spiteri, Rahul Urgaonkar, and Ramesh K. Sitaraman. 2020. BOLA: Near-Optimal Bitrate Adaptation for Online Videos. *IEEE/ACM Transactions on Networking* 28, 4 (2020), 1698–1711.

[62] Julia Stoll. 2019. Number of Paid SVoD Services Subscriptions Among Users in the U.S. https://www.statista.com/statistics/786665/number-paid-svod-service-subscriptions-us/.

[63] Julia Stoll. 2019. TV Binge-Watching Preferences in the U.S. by Age 2019. https://www.statista.com/statistics/687388/binge-watching-preference-usa/.

[64] Makarand Tapaswi, Martin Bauml, and Rainer Stiefelhagen. 2014. StoryGraphs: Visualizing Character Interactions as a Timeline. In *Proc. of IEEE/CVF CVPR*. 827–834.

[65] trillmercy. 2020. Skip Intro Button Has Been Added. https://www.reddit.com/r/Hulu/comments/iqqhi4/skip_intro_button_has_been_added/.

[66] Haoqian Wu, Keyu Chen, Yanan Luo, Ruizhi Qiao, Bo Ren, Haozhe Liu, Weicheng Xie, and Linlin Shen. 2022. Scene Consistency Representation Learning for Video Scene Segmentation. In *Proc. of IEEE/CVF CVPR*. 14001–14010.

[67] Xiao Wu, Alexander G. Hauptmann, and Chong-Wah Ngo. 2007. Practical Elimination of Near-Duplicates from Web Video Search. In *Proc. of ACM MM*. 218–227.

[68] Wen Xia, Hong Jiang, Dan Feng, Fred Douglis, Philip Shilane, Yu Hua, Min Fu, Yucheng Zhang, and Yukun Zhou. 2016. A Comprehensive Study of the Past, Present, and Future of Data Deduplication. In *Proc. of IEEE*, Vol. 104. 1681–1710.
[69] Francis Y Yan, James Hong, Hudson Ayers, Keyi Zhang, Chenzhi Zhu, Philip Levis, Sadjad Fouladi, and Keith Winstein. 2020. Learning in Situ: A Randomized Experiment in Video Streaming. In *Proc. of USENIX NSDI*. 495–511.
[70] Xinlei Yang, Wei Liu, Hao Lin, Zhenhua Li, Feng Qian, Xianlong Wang, Yunhao Liu, and Tianyin Xu. 2023. Visual-Aware Testing and Debugging for Web Performance Optimization. In *Proc. of ACM WWW*. 2948–2959.
[71] Christoph Zauner. 2010. Implementation and Benchmarking of Perceptual Image Hash Functions. (2010).
[72] Haoyu Zhang, Ganesh Ananthanarayanan, Peter Bodik, Matthai Philipose, Paramvir Bahl, and Michael J Freedman. 2017. Live Video Analytics at Scale with Approximation and Delay-Tolerance. In *Proc. of USENIX NSDI*. 377–392.
[73] Bilei Zhu, Wei Li, Zhurong Wang, and Xiangyang Xue. 2010. A Novel Audio Fingerprinting Method Robust to Time Scale Modification and Pitch Shifting. In *Proc. of ACM MM*. 987–990.