Journal of Artificial Intelligence and Technology, 2025, 5, 453-465
https://doi.org/10.37965/jait.2025.0799

RESEARCH ARTICLE

Video Deduplication Using Clustering and Hashing-Based Layered Coarse Resolution Approach for Cloud Storage

Shilpa Chaudhari,[1] R. Aparna,[1,2] Aryan Anchalia,[1] Aneesh M Somayaji,[1] and Anirudh Sanal Kumar[1]
[1]Department of Computer Science and Engineering, M. s. Ramaiah Institute of Technology *(Affiliated VTU)*, Bangalore, India
[2]School of Computer Science and Engineering, RV University, Bangalore, India

(Received 26 April 2025; Revised 17 July 2025; Accepted 19 August 2025; Published online 20 September 2025)

Abstract: Managing storage effectively is crucial in the modern era of growing video data on cloud systems. The exponential increase in video content demands innovative solutions to manage storage space without compromising data integrity and access speed. Video deduplication techniques help to address the issue related to storage efficiency. Existing deduplication approaches either focus on computationally demanding deep learning techniques that limit deployment in resource-constrained situations or employ classical hashing to target precise duplication. This paper integrates clustering and hashing techniques at two resolution layers for video deduplication to maximize cloud-based storage efficiency toward reducing redundant data and improving system performance. Two resolution layers are clustering and hashing. Clustering layer groups similar videos based on video meta-parameters such as frame count and frames per second. Each cluster maintains metaparameters based on its videos. These cluster meta-parameters are compared with meta-parameters of the query video that comes to cloud storage. The matched cluster is considered as flagged cluster. Hash value of each video in the cluster is also maintained. Hashing layer compares each video hash value of the flagged cluster with query video hash value for deduplicate check using hashing-based similarity check logic. This layered resolution approach not only enhances accuracy but also significantly reduces the computational load and time required for deduplication. This approach not only supports scalable and cost-effective video storage solutions but also ensures data integrity and seamless access to multimedia content with improved resource management and user experience.

Keywords: cloud storage; hashing; K-means clustering; video deduplications

I. INTRODUCTION

Low-cost storage services on cloud servers motivates enterprises and individuals to store their data on cloud servers. The exponential expansion of video content in today's digital environment presents serious storage management difficulties, especially for cloud-based systems [25]. Managing this enormous volume of data effectively is crucial to making the most of resources and guaranteeing seamless access to multimedia materials. Effective video content management is essential in the age of expanding digital data and cloud-based storage [1]. There are issues with storage efficiency and resource utilization due to the emergence of duplicate video data. In cloud systems, decreasing storage overhead and improving data organization can be achieved through the process of video deduplication, which involves detecting and eliminating redundant video segments [2]. Numerous methods have been put forth to deal with these issues, including metadata-enabled deduplication and content-driven cache management. The use of these approaches address the growing amounts of video data in cloud storage systems.

Furthermore, cutting-edge techniques for improving storage security and efficiency include edge computing, blockchain-based deduplication, and sophisticated feature extraction algorithms [26–28]. Frame-by-frame video deduplication check is time-consuming which can be replaced with layered coarse resolution approach.

The proposed layered coarse resolution approach aims to address the challenge of managing large-scale video data efficiently on cloud by integrating hashing and clustering techniques in layered coarse resolution approach for video deduplication check. The two layers proposed are not completely independent. Layer 1 is to filter out the videos whose meta-parameters are different from the new video which helps in checking hash values of lesser videos. SHA-265 uses the meta-parameter values for hashing. The goal is to improve storage space utilization, enhance content discovery, and streamline content moderation. To achieve this goal, the system will extract meta-parameters from videos, group similar videos together based on the extracted meta-parameters, and identify and remove duplicates or near-duplicates using the hash values. The video deduplication check is designed using layered coarse approach where two coarse layers based on clustering and hashing respectively are designed. The first step in the layered approach is to extract meta-parameters of video such as frames per second and frame count. Coarse layer-1 groups the similar video segments based on meta-parameters and stores in the database on the cloud. Each video's hash value is computed and stored along with the clusters in the database. Coarse layer-2 compares the cluster video hash value with the input video hash value to identify and remove duplicated content with efficiency, saving a significant amount of storage and enhancing system performance.

Our specific contribution are as follows. (1) Design and develop feature extraction logic for representation of video content in terms of meta-parameters. (2) Design and develop meta-parameters-based clustering technique capable of grouping similar videos together effectively for deduplication check, considering the

Corresponding author: Shilpa Chaudhari (e-mail: shilpasc29@msrit.edu).

extracted meta-parameters. (3) Design and develop hash-based similarity comparison logic using meta-parameters-based clustering for duplicates identification with high precision. (4) Performance analysis of the clustering and hashing-based layered resolution approach for video deduplication.

The remaining paper is organized as follows. Section II discusses the related works in the field of video deduplication using clustering and hashing techniques. Section III presents methodology used for the proposed clustering and hashing-based layered resolution approach for video deduplication. Section IV analyses the obtained results and finally concludes in Section V.

II. RELATED WORKS

Research on video deduplication has increased dramatically as a result of the quick expansion of multimedia data, with an emphasis on safe storage, effective retrieval, and less redundancy in cloud systems. Recent research has investigated more sophisticated approaches to address security, scalability, and near-duplicate detection, even while conventional hashing and clustering algorithms provide lightweight solutions.

Even though data deduplication techniques [29–33] and image deduplications techniques [34–36] exist in literature, this section discusses video deduplication techniques. The authors in [21] address the challenges of video deduplication arising due to the rapid growth of video data. While deduplication is performed by the traditional hashing techniques at the file level using binary hash comparisons, this study introduces an approach of frame-level deduplication. The proposed method first checks for full video duplication using global feature extraction. The video is segmented into frames if no exact match is found, and then deep learning techniques like convolutional neural networks (CNNs) are used to extract over 1000 features per frame. These features are stored and compared using the Euclidean distance measure to check duplication at the frame level. The approach achieves a 95.6% accuracy in detecting duplicate frames, thus outperforming existing state-of-the-art methods [21].

The study in [22] focuses on improving hyperspectral image (HSI) clustering, which is a challenge owing to high-dimensional and complex spectral structures. Conventional subspace clustering methods are designed for a single view, and they do not fully make use of spatial or textural features. To address this problem, the authors propose a contrastive multiview subspace clustering approach which is based on graph convolutional networks. By making use of pixel neighbor textural and spatial–spectral information, this method constructs two graph convolutional subspaces, learning their affinity matrices. The approach was tested on four HSI datasets – Indian Pines, Pavia University, Houston, and Xu Zhou, achieving accuracies of 97.61%, 96.69%, 87.21%, and 97.65%, respectively, thus outperforming existing clustering methods [22].

The authors in [23] address the challenges with respect to referring video object segmentation (RVOS), which is segmenting objects based on textual descriptions. Traditional approaches treat RVOS as a sequence prediction problem, where there is the processing of each frame separately, without quite capturing the inter-frame relationships or temporal variations in object descriptions. To overcome this limitation, the study introduces the Semantic-assisted Object Cluster (SOC) framework, which achieves unified temporal modeling and cross-modal alignment by combining video content and textual guidance. SOC facilitates joint space learning across modalities and time steps, by linking frame-level object embeddings with language tokens. Many experiments on standard RVOS benchmarks show that the proposed method surpasses state-of-the-art approaches, by offering enhanced segmentation stability and adaptability in handling text-based temporal variations [23].

The study in [24] deals with multilevel image thresholding and clustering, which are computationally intensive but popularly used in image processing. To address this challenge, the authors analyze the performance of the Chimp Optimization Algorithm (ChOA) for image segmentation and clustering. The approach makes use of multilevel thresholding to each color channel and implements ChOA to optimize the process effectively. To evaluate the effectiveness of ChOA, several performance metrics were used, including Segment Evolution Function, Peak Signal-to-Noise Ratio, Structural Similarity Index, and Probability Rand Index. Its performance was compared with eight well-known metaheuristic algorithms, such as Particle Swarm Optimization, Whale Optimization, and Grey Wolf Optimization, using Kapur's entropy and Otsu's class variance methods. The results indicate the competitive performance of ChOA, achieved in image segmentation and clustering [24].

As authors in [2] state, data deduplication is essential for reducing redundancy and compressing data, particularly in cloud storage. By encrypting content before storage, the Efficient Hash Function-based Duplication Detection (EHFDD) technique described in this research improves data security in cloud contexts. It delivers enhancements for verified duplicate verification in hybrid clouds and tackles issues with current deduplication techniques. When compared to existing methods, EHFDD minimizes overhead, resulting in notable improvements: a reduction of 28.7 milliseconds in average delay, 8% less memory utilization, a reduction of 457 milliseconds in computation time, a reduction of 900 milliseconds in communications overhead, and a rise of 3.13% in success rate [2].

The study by [3] essentially describes how to use the Distributed Storage Hash Algorithm (DSHA) to remove duplicate data from the cloud. Conventional hash algorithms store the output in a fixed length of 128- or 160-bit memory, plus an additional amount of memory for storing the hash value. The DSHA algorithm that is being utilized here is far superior to the standard MD5 or other secure hashing algorithms since it provides greater read/write performance and requires less memory space for the hash value to be stored [3].

FastCDC, an effective content-defined chunking (CDC) method for data deduplication systems, is presented in [4]. Five essential strategies are used by FastCDC to produce a 3–12X speed increase over current CDC methods while retaining comparable or greater deduplication ratios. These strategies include a gear-based rolling hash and optimal judgment. Furthermore, FastCDC outperforms existing chunkers by 1.2–3.0X in deduplication efforts like Destor [4].

The authors of [5] describe how employing specific content-driven cache management techniques can enhance the performance of deduplication storage. The current deduplication-aware caching methods function properly when the cached block size is 4 KB. However, when the block size is adjusted to be larger than 4 KB, performance is negatively impacted. Because of the varying read and write alignment, this also results in incredibly low cache space utilization. In order to solve these issues and provide better results, CDAC based on LRU and ARC algorithms, or CDAC-LRU and CDAC-ARC, respectively, should be used [5].

The paper [6] presents Classified-Metadata-based Restoring (CMR), a solution to deduplication system fragmentation. CMR aggressively prefetches chunk metadata and optimizes memory utilization by classifying backup metadata into files and chunks. By lowering disk reads, boosting throughput without compromising deduplication ratio, and effectively utilizing hardware resources, this technique enhances restoration performance. CMR improves deduplication ratio by 1.91% and 4.36% while reducing restoring time by 27.2% and 29.3%, respectively, according to comparative trials versus history-aware and context-based rewriting approaches. CMR helps deduplication systems operate better and use less storage space [6].

PM-Dedup, a safe deduplication framework that uses trusted execution environments (TEEs) to carry out partial verification at the edge, was suggested by Ke et al. in 2025. This method protects user privacy while relieving cloud servers of some of their computing load. PM-Dedup securely authenticates and deduplicates data blocks via a key-sharing approach between edge and cloud nodes. Without sacrificing data confidentiality, the authors show notable gains in system throughput and latency. According to their findings, in hybrid cloud environments, deduplication efficiency is 40% higher than with traditional encrypted deduplication techniques [7].

Wu and Fu (2024) used Intel SGX-based enclaves to develop a secure metadata deduplication architecture for cloud-edge systems. This technique preserves the security of metadata while enabling cooperative deduplication between edge devices and cloud servers. Using trusted enclaves, the system streamlines key management and minimizes redundant data transmission. In comparison to conventional cloud-only techniques, their performance investigation on real-world workloads shows a 30–50% decrease in deduplication overhead and up to 60% faster processing time [8].

A hybrid deduplication strategy tailored for edge computing environments was created by Shin et al. in 2022. The method uses symmetric encryption techniques to ensure security while supporting both client-side and server-side deduplication. By using local caching and edge-to-cloud metadata synchronization, it reduces communication and storage overhead. Their experiments on IoT and video datasets show lower latency and better deduplication ratios. Additionally, by limiting data availability to unauthorized parties, the study addresses privacy threats [9].

DEBE (Deduplication Before Encryption), a safe deduplication protocol, was presented by Yang, Li, and Lee (2022). It uses Intel SGX enclaves to deduplicate data before encryption. DEBE permits similarity verification without disclosing data content, in contrast to conventional techniques where encryption prevents deduplication. By reducing duplicate storage and maintaining security, the approach improves performance. According to evaluations, DEBE outperforms conventional encryption-first techniques in terms of performance by up to 1.6× and achieves high deduplication accuracy [10].

The concept of a secure block-level deduplication solution for cloud data centers is given in [11]. However, there is a lot of data waste. For example, when someone opens some data and uploads it to a drive, the space will be wasted if someone else uploads the same identical data. This is why data redundancy is a major problem. After testing the two deduplication techniques on a local dataset, the authors compared the widely used file-level deduplication with their block-level deduplication for cloud data centers and discovered that the block-level deduplication approach yields 5% better results than the file-level deduplication approach [11].

A safe fuzzy deduplication system for multimedia and near-duplicate data scenarios was created by Tang et al. in 2023. To guarantee semantic similarity identification before to real deduplication, the system incorporates a label consistency pre-verification phase. By preserving content integrity and thwarting hostile manipulations, it facilitates encrypted environments. According to their findings, the approach is appropriate for surveillance and instructional archives since it can accurately identify 85–90% of almost identical videos in safe environments [12].

A thorough analysis of video fingerprinting algorithms, ranging from early frame-level hashing to more recent perceptual and deep learning-based approaches, is provided by Allouche and Mitrea (2022). In order to objectively analyze robustness, computational cost, and resistance against video modifications (such as scaling, re-encoding, and cropping), they divide techniques into three categories: spatial, temporal, and hybrid fingerprints. The authors draw attention to the growing trend of perceptual hashing models, which are appropriate for large-scale multimedia systems because they strike a compromise between detection accuracy and real-time speed. As a foundational reference for research in effective duplicate or near-duplicate video detection, the paper ends by outlining future approaches including cross-modal fingerprinting and lightweight neural networks [13].

To index high-dimensional vectors and avoid expensive retraining, it has an effective ANNS layer. Short CNN and ORB features maximize deduplication, while auditory characteristics improve identification [14]. The issues of complexity and scalability are discussed. It finds duplicates precisely using its clip-based matching technique. Almost 800k videos are indexed every day by Maze, which has been in use for two years. 4.84 s write latency and 4 seconds read delay are displayed by the ANNS layer. Reduced data migration because there is no need for retraining. Maze recognizes comparable live streaming videos both visually and sonically, with 98% recall. With only 250K standard cores needed for every billion films, it is incredibly economical, saving 5800 SSDs [14].

An effective cloud storage structure for GOP (Group of Pictures) level deduplication with adaptive GOP structure is discussed in [1]. For the purpose of optimizing storage utilization, cloud storage systems use deduplication. A critical component, GOP-level deduplication, makes video deduplication possible. To improve deduplication efficiency over fixed-size GOP structures (e.g., 8, 10, 12, 15), we suggest an adaptive GOP structure. With a 2.18% PSNR improvement, our approach outperforms GOP structures with fixed sizes. Duplicate frame detection is improved by adaptive GOPs, which guarantee closer relationships between frames. By providing a viable method for effective deduplication, this study tackles the problem of memory wastage in cloud storage systems, especially for video data [1].

The difficulties in implementing video deduplication in cloud storage systems is discussed in [15]. Both individuals and enterprises can benefit from cloud storage since it makes remote data access and storage possible over the internet. But sharing storage makes it more likely that video data will be duplicated, which wastes a lot of storage space. A solution is deduplication; however, because video storage is so complicated, it might be difficult to execute video deduplication efficiently. The goal of this research is to pinpoint these issues and provide solutions, illuminating efficient methods for video deduplication in cloud storage settings [15].

The technique of video deduplication utilizing CNNs and SHA-256 algorithms is discussed in [16]. With the exponential growth of digital data output, customers are turning to cloud service providers to store large amounts of video footage, which can be expensive. Thus, it is essential to use efficient methods to minimize storage costs. With an emphasis on extracting both global and local information of films, this study provides a comprehensive analysis of video deduplication systems. The use of hashing algorithms like SHA-256 for global feature extraction and Conv2d (CNN) algorithms for local feature extraction are important strategies. These strategies are essential for cutting down on redundancy and maximizing storage use in cloud computing settings [16].

The authors of [17] discuss the use of private media sharing and secure deduplication in mobile cloud computing. In this study, we propose SMACD, a multidimensional media sharing system for mobile cloud computing that preserves anonymity. It encrypts each media layer with fine-grained access control using attribute-based encryption. Hierarchical access control is ensured by building multilevel access policies with secret sharing. Both intra- and inter-server deduplication are facilitated by decentralized key servers. Test results show that SMACD effectively preserves media privacy at low computational and storage costs.

The secure image deduplication system presented in this research combines clustering and hashing methods with a unique perceptual hash algorithm based on Local Binary Pattern analysis. In this approach, image transmission takes place in encrypted form, and image hash values act as distinct fingerprints for deduplication. First, the images in the image set are pre-clustered on the cloud storage server. For an image, the hash value of the image is calculated by the LBP-based perceptual hash (LBPH) algorithm. The client calculates the perceptual hash values of images and encrypts the images using the symmetric encryption algorithm, before both being uploaded to the cloud server. After the user uploads the image hash, the server performs the detection of duplicates as well as image clustering. It uses the similarity of fingerprints and the comparison of Hamming distances to perform the deduplication in an encrypted form, protecting the confidentiality of the images. Image retrieval involves locating the image class and matching fingerprints, with the encrypted images being returned to users for decryption using their private keys. Based on its comparative performance against other image deduplication algorithms, this model guarantees image security, also improving deduplication accuracy [18].

This proposed model presents a unique method called SCDS (Similarity Clustering-based Deduplication Strategy), which is intended to reduce system resource consumption and maximize the efficiency of duplicate data removal. The SCDS method optimizes fingerprint index querying by utilizing similarity clustering and data partitioning methods. A partitioning algorithm is first used to preprocess the data in order to group together similar data. A similarity clustering algorithm is then used to divide the fingerprint superblock into clusters of similar data during the data elimination phase. Repeated fingerprints within the cluster are identified to speed up the retrieval of duplicate fingerprints. Experimental results demonstrate that SCDS outperforms existing similarity deduplication methods in terms of deduplication ratio, although with only a marginal increase in overhead compared to certain high-throughput but low-deduplication ratio techniques [19].

The challenge of handling enormous amounts of digital data is examined in this research [20], which raises performance demands, backup costs, and storage capacity issues. The inability of traditional backup systems to stop data duplication during backups results in longer backup times and wasteful resource usage. By effectively removing redundant data, data deduplication reduces storage usage and provides a solution. Its goal is to swiftly and efficiently accomplish the best possible duplicate removal while balancing large deduplication ratios with short backup windows. It studies and categorizes current deduplication techniques, assesses their performance indicators, and suggests a novel way to maximize deduplication ratios with the least amount of backup windows. It also highlights important questions for additional study [20].

There is still a considerable gap in obtaining scalable, privacy-preserving, and near-duplicate-aware deduplication that is suited for real-time cloud storage systems, even with the notable advancements in video deduplication techniques, such as frame-level analysis, CNN-based approaches, perceptual hashing, and hybrid cloud-edge frameworks. The majority of current approaches either concentrate on computationally demanding deep learning techniques that restrict deployment in resource-constrained situations or use classical hashing to target precise duplication. Moreover, fuzzy duplicates videos with slight temporal or visual changes are poorly handled by existing systems, especially in encrypted or dispersed storage environments. Although some recent research suggest label consistency verification or secure deduplication utilizing TEEs, they are not flexible enough for varied data formats and large-scale streaming material. Furthermore, very few methods make an effort to combine security, flexibility, and efficiency in a layered deduplication framework. A scalable, resource-efficient, and secure layered deduplication system that can handle both exact and near-duplicate films intelligently is therefore desperately needed, especially in contemporary cloud-edge systems.

Secure video deduplication is considered in [2,3,7,11, 13,17,20], while [1,4-6,8-10,12,14-16,18,19] do not provide security during the deduplication process. These techniques are compared in Table I in terms of study area, parameters used for deduplication process, dataset used, and techniques used.

III. PROPOSED VIDEO DEDUPLICATION USING CLUSTERING FOR EFFICIENT DATA STORAGE ON CLOUD

Video deduplication in cloud storage systems is active and diverse, incorporating a variety of cutting-edge methods and approaches. Researchers are always looking at new ways to increase storage security and efficiency, from edge computing techniques to distributed storage algorithms. Although there are still issues with scalability and metadata management, the proposed video deduplication presents encouraging prospects aiming to reduce storage management time and improve cloud service performance for high-definition video information. Its sophisticated technology takes a multipronged approach, beginning with a careful examination of incoming video files to understand the subtleties of their content via meta-parameter calculation.

The architecture of the proposed video deduplication consists of two coarse resolution layers – coarse layer-1 called as meta-parameters-based K-means clustering and coarse layer-2 called as hash-based similarity comparison logic as shown in Figure 1. Coarse layer-1 involves computation of video meta-parameters extraction and K-means-based video clustering. Coarse layer-2 involves hash-based duplicate check and updating of video storage on cloud.

Computation of video meta-parameters extraction creates the foundation for finding commonalities with content already in the database. It makes it easier to find the closest video cluster quickly. The process starts when a user uploads a new video, called a “query video,” which sets off a complex chain of events meant to guarantee data integrity and storage optimization. After upload, the system calculates the query video’s meta-parameters, which

Table I. Comparison of video deduplication techniques

Paper	Area of study	Parameters used	Datasets	Techniques used
[1]	Cloud storage	Deduplication efficiency, GOP structures, PSNR improvement	Video- non-complex/complex textured, low/medium intensity	Adaptive GOP structure, GOP-level deduplication
[2]	Image deduplication in cloud storage	Hamming distance, clustering threshold, deduplication threshold	IVC-LAR database (8 original color images, 120 distorted images)	LBPH, K-means clustering, symmetric encryption
[3]	Cloud data centers	Data redundancy, storage efficiency, backup systems improvement	VIDEO, UQ_VIDEO, CC_WEB_VIDEO, YouTube videos	Block-level deduplication, Bloom filter, Index classification
[4]	Mobile cloud computing	Media sharing, privacy preservation, secure deduplication, fine-grained access control	Mobile datasets, Surrey University Library for Forensic Analysis (SULFA), manually recorded videos	Attribute-based encryption, hierarchical access control, decentralized servers
[5]	Non-shared storage systems	Content fingerprints, metadata management, storage efficiency	CINFINITY, TRECVID2010, VCDB, CC_WEB_VIDEO	Content-defined chunking, content addressable deduplication
[6]	Deduplication storage systems	Cache management, performance enhancement, block size optimization	Traffic video sequence, CCTV_WEB_VIDEO, CC_WEB_VIDEO	Content-driven cache management, LRU and ARC algorithms
[7]	Video deduplication	Redundancy reduction, storage maximization	FIVR-200K dataset, CDB (video copy database), MCL-ONEVID dataset	CNN, SHA-256 hashing
[8]	Encrypted deduplication storage	Metadata management, storage efficiency, security assurances	Dataset – Todou/Vimeo/MCL-ONEVID; Video – YouTube, non-complex/complex textured, low/medium intensity	Key distribution strategy, integration of servers
[9]	Backup systems	Memory consumption, disk accesses, backup throughput	UQ_VIDEO, CINFINITY, surveillance videos, SULFA	Bloom filter, Index classification, memory optimization
[10]	Non-shared storage systems	Content fingerprints, metadata management, storage scalability	Dataset – Todou/Vimeo/MCL-ONEVID, Video – YouTube	Inline deduplication, chunking
[11]	Decentralized storage systems	Bandwidth reduction, security assurances	Dataset – Todou/Vimeo/MCL-ONEVID; Video – YouTube	Ganache, Ethereum blockchain
[12]	Cloud data centers	Data redundancy, storage efficiency, backup systems improvement	SULFA, manually recorded videos, CCTV_WEB_VIDEO	Block-level deduplication, file-level deduplication
[13]	Cloud storage systems	Random access patterns, fast response times, metadata overhead	Traffic video sequence, CCTV_WEB_VIDEO, CC_WEB_VIDEO	Block-based partial deduplication, similarity-based indexing
[14]	Edge computing	Data privacy, response times, communication costs	SULFA, manually recorded videos, CCTV_WEB_VIDEO	Client-side and server-side approaches
[15]	High-dimensional vector indexing	Duplicate detection, resource consumption, scalability	FIVR-200K dataset, CDB (video copy database), MCL-ONEVID dataset	ANNS layer, clip-based matching technique
[16]	Video deduplication	Video storage complexity, deduplication efficiency	CCTV_WEB_VIDEO, CC_WEB_VIDEO	Not specified
[17]	Video deduplication	Redundancy reduction, storage maximization	CDB (video copy database), MCL-ONEVID dataset	CNN, SHA-256 hashing
[18]	Mobile cloud computing	Media sharing, privacy preservation, secure deduplication	Mobile datasets, SULFA	Clustering, perceptual hash algorithm, secret sharing
[19]	Duplicate data removal	Deduplication ratio, resource consumption, overhead	YouTube videos, Vimeo, Todou dataset	Similarity clustering, data partitioning
[20]	Cloud storage	Data redundancy, backup efficiency, storage capacity	UQ_VIDEO, CC_WEB_VIDEO	Not specified
[21]	Video deduplication	Deduplication accuracy, frame-level feature extraction, Euclidean distance measure	Public Videos Dataset, YouTube-8M	CNN-based feature extraction, local feature comparison, Euclidean distance
[22]	Hyperspectral image clustering	Clustering accuracy, spatial–spectral feature extraction, contrastive learning	Indian Pines, Pavia University, Houston, Xu Zhou	Graph convolutional networks, multiview subspace clustering, contrastive learning, attention-based fusion

(continued)

Table I. *(continued)*

Paper	Area of study	Parameters used	Datasets	Techniques used
[23]	Referring video object segmentation (RVOS)	Temporal coherence, cross-modal alignment, segmentation accuracy	DAVIS Video Segmentation Dataset, YouTube-VOS	Semantic-assisted object cluster, frame-level object embedding, multi-modal contrastive supervision
[24]	Image clustering and segmentation	Thresholding efficiency, image quality assessment, segmentation accuracy	BSDS500-Image Segmentation Data-set, COCO Dataset, PASCAL VOC Dataset	Chimp Optimization Algorithm (ChOA), multilevel thresholding, Kapur's entropy method, Otsu's class variance method

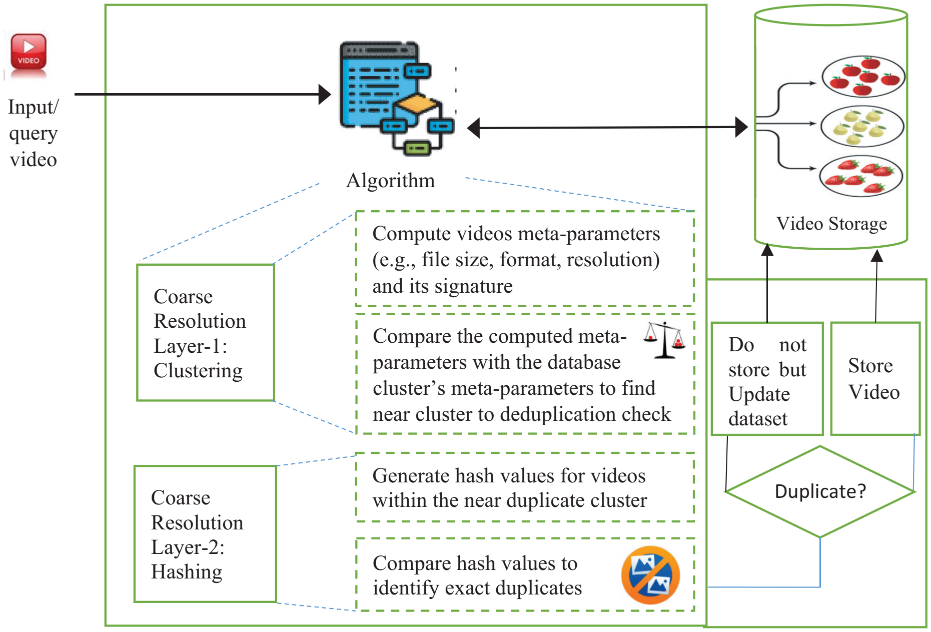
Input/ query video
Algorithm
Video Storage
Compute videos meta-parameters (e.g., file size, format, resolution) and its signature
Coarse Resolution Layer-1: Clustering
Compare the computed meta-parameters with the database cluster's meta-parameters to find near cluster to deduplication check
Generate hash values for videos within the near duplicate cluster
Coarse Resolution Layer-2: Hashing
Compare hash values to identify exact duplicates
Do not store but Update dataset
Store Video
Duplicate?

Fig. 1. Proposed video deduplication.

include a variety of extracted meta-parameters like the length of the video and frames per second of the video.

Together, these meta-parameters show a distinct signature for the video, which is necessary for later comparisons and duplication detection. The hash value is also computed for each video. The videos are clustered based on the extracted meta-parameters using K-means clustering. All the clusters are stored in video dataset along with meta-parameters and hash value. The decision of adding the query video in the cluster is based on the hash-based similarity logic. If the hash of query video matched with any hash in the any cluster, then it is declared as duplicate video. When a duplicate query is found, the system cleverly modifies the database to only point to the updated query, eliminating the need for physical storage and optimizing database usage. This is done for the retrieval of the video in future. In addition to increasing storage efficiency, this deduplication technique yields observable advantages, including lower costs and faster retrieval times, reaffirming its role as a vital component for cloud-based data management systems aiming for optimal efficiency.

This sequence diagram in Figure 2 depicts how a video deduplication and storage algorithm interact with the user, algorithm, and database. The procedure begins with the user uploading a movie, after which the system extracts its meta-parameters, which include resolution, frame rate, duration, and keyframe signatures. These meta-parameters are then compared to those of video clusters already in the database. The system retrieves the cluster to which the input video may belong and its accompanying hash values. The system detects whether the input video is unique or duplicate by comparing its extracted meta-parameters to the recovered cluster data.

If the video is unique, it is saved in the database, and the user is notified when it is successfully stored. However, if the system identifies the video as a duplicate, it is not stored again. Instead, it saves a reference to the existing duplicate movie and tells the user. This approach enables effective storage utilization by minimizing redundancy, which saves disk space and lowers data management overhead. Furthermore, grouping movies based on meta-parameters makes retrieval faster and improves system performance. This technology, which automates the deduplication process, dramatically improves storage economy and scalability, making it ideal for large-scale video storage and retrieval.

A. *COARSE LAYER-1: VIDEO META-PARAMETERS-BASED K-MEANS CLUSTERING*

Video meta-parameters considered in the proposed approach includes its duration, frame rate, and resolution. These meta-parameters are essential for comprehending the qualities of the video and serving as the foundation for clustering. Developers employ a range of techniques to extract video meta-parameters. These video meta-parameters offer crucial details about the visual and structural elements of the video content, laying the groundwork for efficient

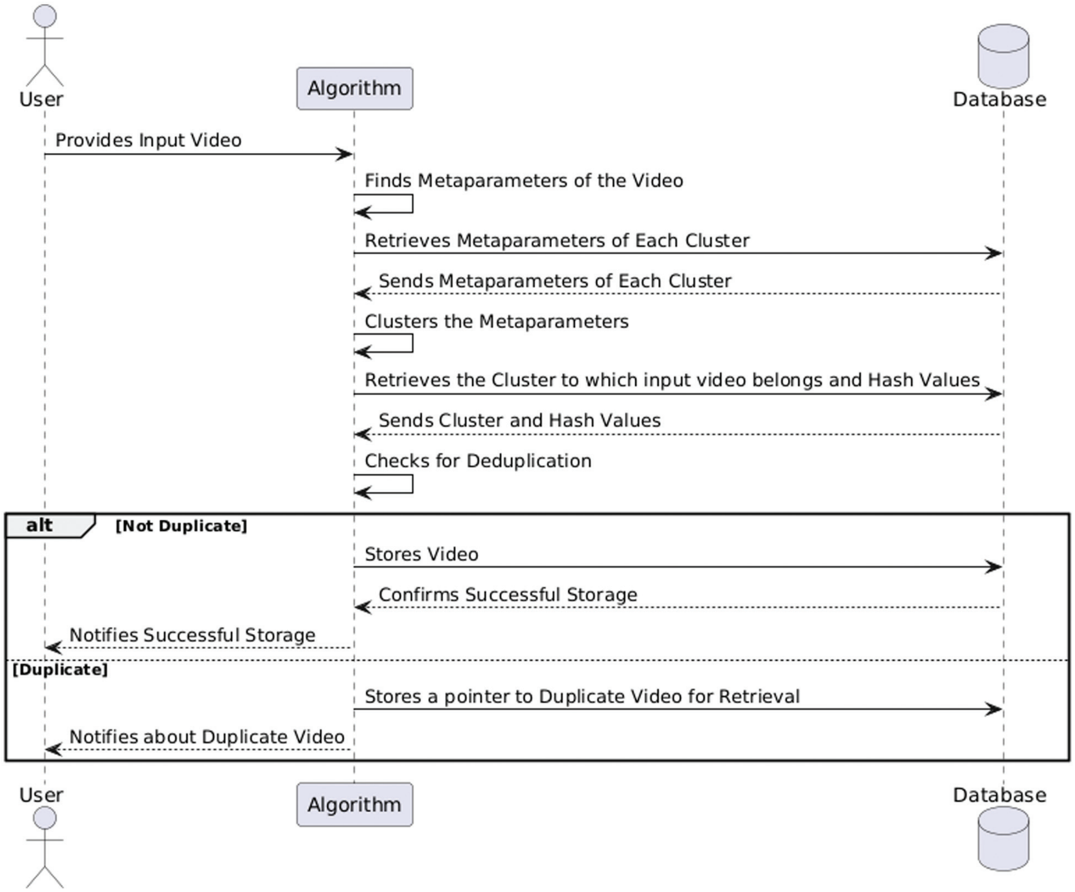
User
Algorithm
Database
Provides Input Video
Finds Metaparameters of the Video
Retrieves Metaparameters of Each Cluster
Sends Metaparameters of Each Cluster
Clusters the Metaparameters
Retrieves the Cluster to which input video belongs and Hash Values
Sends Cluster and Hash Values
Checks for Deduplication
alt [Not Duplicate]
Stores Video
Confirms Successful Storage
Notifies Successful Storage
[Duplicate]
Stores a pointer to Duplicate Video for Retrieval
Notifies about Duplicate Video
User
Algorithm
Database

Fig. 2. *S*teps of sequence in video deduplication system.

processing and analysis. Using Python's OpenCV package, the project calculates the length of the video and frames per second (fps). Video length and fps provide a basic understanding of the video's temporal structure and playback quality, making them essential factors for clustering comparable videos in the proposed video deduplication checks. The fps gives information about motion smoothness and quality, which is crucial for differentiating between videos with varying frame rates. Video length aids in determining the overall time and ensures that comparisons are made between videos of similar lengths. Together, these two meta-parameters guarantee that videos are compared on the basis of both their content and structural features, which facilitates precise grouping and effective deduplication. Video length and frames per second (fps) are essentially constant across various encodings and codecs, and they are more beneficial than other meta-parameters for video deduplication. Length and frames per second are powerful markers of video identity, in contrast to resolution or bitrate, which might change depending on compression settings. Before conducting a more thorough investigation, these characteristics offer a simple yet efficient method of grouping possible duplicates. Re-encoding can cause large changes in file size and codec types, which makes them unreliable for deduplication. Additionally, scalability is improved by length and fps, which effectively narrow the search space for additional comparisons. These metrics are still robust for clustering because little changes like watermarking or light cropping do not dramatically change fps and length.

Consider Vq as the query video notation located at $Video\_path$ and Mq as the meta-parameters of Vq. The meta-parameter is computed as a function of video that computes the meta-parameters, that is, $Mq = f(Vq)$. Meta-parameters include video length $L(Vq)$ and fps. The number of frames are computed as $frame\_count$ in Vq. *VideoCapture* object of the *OpenCV* library (*cv2*) is used to extract various properties and frames from the video. The $frame\_count$ and fps in Vq are retrieved using *get(cv2.CAP_PROP_FRAME_COUNT) and get(cv2.CAP_PROP_FPS)*, respectively. The length of the Vq is computed as shown in Equation 1. The meta-parameters are stored in Mq as shown in Equation 2:

$$L(Vq) = \frac{frame\_count}{fps} \tag{1}$$

$$Mq = \{L(Vq), fps\} \tag{2}$$

The *get(cv2.CAP_PROP_FRAME_WIDTH)* and *get(cv2.CAP_PROP_FRAME_HEIGHT)* retrieve the width and height of each video frame in Vq, respectively. These values represent the resolution of the video. The $meta - parameters$ are stored in the video database along with the actual video data. The strong database infrastructure, which acts as the framework for storing and organizing meta-parameters, clustering data, and video data, is essential to the entire process. It adds a great deal to the overall efficacy and efficiency of the deduplication process by guaranteeing the smooth storage of non-duplicate movies and facilitating the efficient retrieval of clusters for comparison. A MongoDB database is used to store and retrieve video metadata and cluster information. A mongodb (NoSQL) database with two key value pair is used – “metadata” and “cluster.” The “metadata” key value pair takes two values: one is the length of the video and the other is the fps of the cluster. The “cluster” key value pair is {video_name: [path_of_the_video, hash_value]} for each video.

The clustering algorithm put videos into different clusters based on similarities based on the video $meta - parameters$ using methods like hierarchical clustering and K-means clustering. By putting videos with similar properties together, this clustering technique makes it easier to identify duplicates, improving data management and storage effectiveness. These $meta - parameters$ are subjected to K-means clustering using Python’s sklearn package. Each $meta - parameter$ in the K-means clustering algorithm is given a weight that indicates how important it is to differentiate between different films. The weights applied to the video length and frame rate are 0.7 and 0.3, respectively. The higher the weight, the more influence that feature will have on the clustering process. Testing has shown that this specific combination of weights works best for clustering comparable movies because it highlights the time of the video, which is frequently a more important aspect in separating content. The weights assigned varies with the video application scenario. Given the significant role that weights play in distinguishing films, the assignment of weights guarantees that the clustering algorithm gives video duration a higher priority. Nevertheless, fps is also thought to record changes in playback quality, which can be useful in differentiating between recordings of comparable lengths but with differing playback qualities. Because length and frame rate are taken into account simultaneously, clustering is balanced and resistant to changes in the quality and substance of the videos. K-means was chosen for clustering in video deduplication using length and FPS because it is efficient, scalable, and well suited for numerical data. Other clustering algorithms were not used due to various limitations. Hierarchical clustering is computationally expensive (O(n^2) or worse) and impractical for large datasets, especially when dealing with continuous features like length and FPS. DBSCAN, while effective for density-based clustering, requires careful tuning of the distance threshold (epsilon), which is difficult with numerical metadata, and struggles with varying cluster densities

The proposed technique maintains a constant number of clusters. This choice is based on the requirement to determine which cluster the input video belongs to while in order to facilitate deduplication, classifying the remaining cluster as redundant. The procedure becomes more focused and streamlined when the number of clusters is kept constant, making it possible to identify significant clusters quickly. The primary objective of clustering is to efficiently organize videos rather than to quantitatively evaluate the clusters. Since the clustering step is not intended for classification, decision-making, or predictive modeling, the use of formal cluster evaluation metrics is not necessary. The approach focuses on grouping similar videos based on metadata parameters such as length and fps to facilitate better storage and retrieval. The effectiveness of clustering is indirectly reflected in the streamlined organization of videos, improved deduplication processes, and reduced computational overhead for further analysis.

Let’s represent $\{C1,C2\ldots,Cn\}$ as the clusters in the database, and let M_{Ci} denote the $meta - parameters$ of i^{th} cluster, Ci. Compare the M_q to each M_{Ci} for finding the flagged cluster for deduplication check. Consider the flagged cluster is denoted as C_f and has m video, which is the target for checking video deduplication making sure that only the most pertinent videos are compared to the input video. By carefully assigning weights to K-means clustering, this methodical approach allows for the precise and efficient retrieval of flagged clusters. This improves the accuracy and efficiency of finding duplicate or extremely similar films by guaranteeing that comparisons are conducted with videos that share comparable structural and playback properties. This improves the deduplication process.

B. *COARSE LAYER-2: HASHING FOR VIDEO DEDUPLICATION CHECK*

The hash values for every video in the flagged cluster are retrieved from the video database. As distinct fingerprints created from the content of every video, hash values make sure that even the smallest variations in one video produce a different hash value. We can quickly ascertain whether the video is already there in the database by comparing the input video’s hash value to those that are already there. The video is unique if the hash value does not exist in the database. If this is the case, we then store the query video in the relevant cluster together with its name and hash value. After that, the database is updated to reflect the query video, guaranteeing that the distinct video and the metadata that goes with it are appropriately indexed for future use. In this case, we preserve efficiency and conserve storage by only saving the name and route to the already-existing movie in the database, as opposed to storing the duplicate video again. This reference makes it simple to access and retrieve the previously saved video without requiring redundancy. This method keeps the database efficient and well organized while also greatly optimizing storage management. The system can more efficiently handle a higher volume of video data by preventing needless duplication. The systematic deduplication procedure also keeps the video management system scalable and efficient while guaranteeing that users can easily locate and access unique recordings. This strong strategy guarantees that the system for storing and retrieving videos can manage large and varied video datasets with ease, improving its overall effectiveness and dependability.

The SHA-256 algorithm, a strong cryptographic hash function that generates a fixed-size 256-bit hash result, is used for hashing. GOP-level hashing, which stands for Group of Pictures, is used. A GOP is a group of consecutive frames that are compressed as a single unit in video compression. GOP-level hashing uses the SHA-256 technique to hash each individual frame of the video. After the hash value for each frame is produced, these values are added together to provide a composite hash value that represents the whole video. With this technique, the temporal sequence and visual data within the GOPs are captured, giving the video content a robust and detailed fingerprint. GOP-level hashing offers the advantage of being more accurate in recognizing and differentiating video footage since it accounts for the subtleties and differences in individual frames. It also makes sure that even slight variations in the video frames produce a distinct hash value, which improves the deduplication process by precisely recognizing unique films and preventing false positives in deduplication checks. A high degree of precision in video content management and retrieval systems is ensured by this thorough hashing technique.

Consider the hash value of the video $H_q = h(V_q)$ where h is the hash function such as using SHA256. Retrieve the hash values $\{H_{Cj}^1, H_{Cj}^2, H_{Cj}^m\}$ of all videos in the flagged cluster C_f. Compare and check the availability of H_q in the retrieved hash values $\{H_{Cj}^1, H_{Cj}^2, H_{Cj}^m\}$. If $H_q \in \{H_{Cj}^1, H_{Cj}^2, H_{Cj}^m\}$, do not store the whole V_q but store the V_q name, V_q path, and M_q in the video dataset on cloud for future retrieval of video. Otherwise store the V_q, V_q name, and V_q path as part of C_f in video dataset and update the cluster with its meta-parameters, M_{Cf} based on M_q. The deduplication check is followed by an outcome-based decision-making procedure within the system. In the event that the check reveals duplication, the system takes a calculated approach to save storage by only updating the database to point to the already-existing video, preventing the need for redundant storage. On the other hand, the system stores the video in the database for later access and retrieval if the check determines that it is unique.

IV. RESULTS ANALYSIS

The project leverages several key tools to achieve its goals. *FFmpeg* is used for extracting meta-parameters, video processing, and frame extraction, offering comprehensive multimedia capabilities. *OpenCV* handles video frames and facilitates frame-by-frame processing with its extensive computer vision algorithms. *Python* is the primary programming language, known for its readability and ease of use. *Numpy* is essential for numerical operations and handling large arrays, while *Pandas* provides powerful data manipulation and analysis capabilities. *Scikit-learn* is employed for implementing clustering algorithms like *K-means*, and *Hashlib* is used for generating *SHA-256* hash values. *Pymongo* facilitates interaction with *MongoDB*, where video metadata, hash values, and clusters are stored and managed.

Execution time is one of the performance parameters for video deduplicate check. Two hashing techniques are used here – one is the proposed hashing based on SHA256 which is GOP-based hash value computation. And the existing model hashing is SHA256 without GOP-based hash value computation. The time taken with each hashing method is shown in Figure 3 to identify and remove duplicate videos from the database. It is highlighting the superior performance of the proposed hashing technique SHA-256, over the hashing algorithm used by the existing model. The chart indicates that the optimized method significantly reduces execution time while comparing it with the traditional approach, and this makes it handier for handling large amounts of video data in cloud storage systems. In real-world applications, this reduced execution time is essential for preserving high throughput and reducing processing delays [2].

Figure 4 compares memory utilization between the proposed and existing approaches in the context of video deduplication. The primary focus of this comparison is on the storage efficiency of hash values generated by the two approaches. It demonstrates how the suggested approach, which uses SHA-256 hashing, takes less memory than the current paradigm. Hash values serve as unique identifiers for video data, so optimizing their storage is critical for improving overall system performance. The results show that the hash size remains consistent across various image sizes in both approaches, implying that the processing needs for hashing are unaffected by the input file dimensions [2].

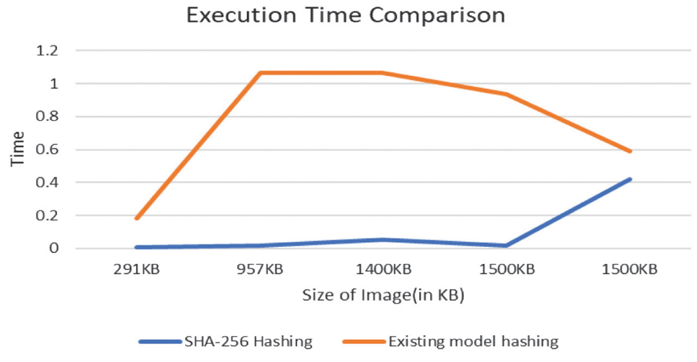

Fig. 3. Execution time comparison.

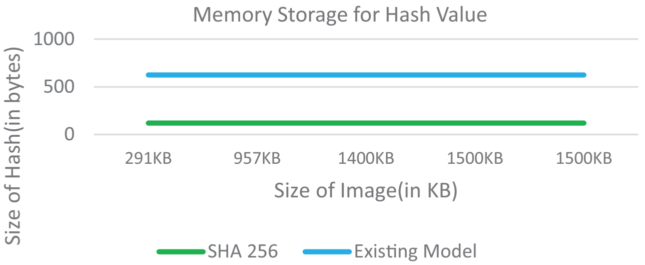

Fig. 4. Memory storage for hash value.

The suggested model's lower memory consumption for SHA-256 contributes significantly to the scalability of deduplication systems, particularly when dealing with big video collections. Video files are typically resource-intensive, needing significant storage and computational power to process. By lowering the amount of memory required to hold hash values, the suggested approach contributes to lower total storage costs and improved resource allocation. This is especially useful in cloud-based or large-scale distributed storage setups, where memory efficiency directly influences performance and cost-effectiveness.

Furthermore, SHA-256's effectiveness in keeping a fixed hash size maintains metadata consistency, making it easier to track and retrieve video files with minimal overhead. The suggested model's capacity to manage rising amounts of data without increasing storage requirements makes it more appropriate for real-world applications like video streaming services, surveillance systems, and digital archives. The findings in Figure 4 highlight the suggested method's advantages in terms of cost-effectiveness, scalability, and optimal memory consumption, making it an excellent alternative for modern deduplication frameworks.

The proposed methodology does not involve training or learning, and it is only required to test its correctness on video samples. The primary focus is on verifying whether the clustering method accurately groups similar videos based on the selected metadata parameters. Therefore, a small sample size was sufficient to demonstrate the algorithm's functionality. The selection of videos was made to cover different variations relevant to the clustering approach, ensuring that the method performs as expected.

Three cases are considered while taking the performance – best case, average case, and worst case. Best-case scenario consists of minimum comparison for deduplication check. Here, the V_q is duplicate in the initial check of the process. Eg. First cluster is C_f and first video in C_f is the duplicate of V_q. Average-case scenario involves average number of comparison for deduplication check. Here, the V_q is duplicate in the middle cluster of the process. For example, middle cluster is C_f and middle video in C_f is the duplicate of V_q. Worst-case scenario involves maximum number of comparison for deduplicate check. Here, the V_q is duplicate in the last cluster of the process. For example, last cluster is C_f and last video in C_f is the duplicate of V_q.

Figure 5 shows the performance of the deduplication algorithm when it is handling maximum number of duplicate videos. Hence, all the videos inputted to the system in this case are

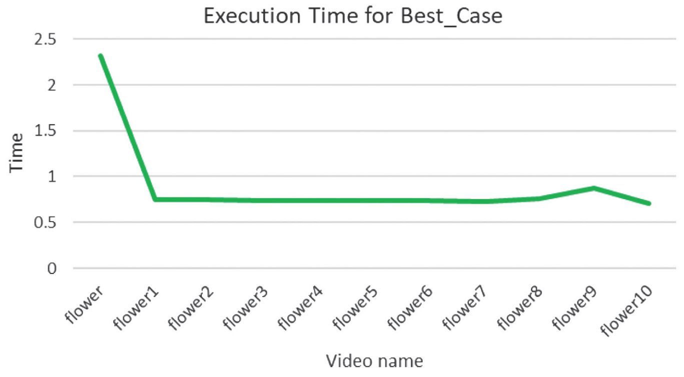

Fig. 5. Execution time for Best_Case.

redundant copies. The graph starts off with a high execution time for the first video, because it is not a duplicate to any at this point, as a result which it is stored in the database. The rest of the query videos being duplicates of the first are not stored to the database, and hence the time taken is less. This execution time required for each duplicate video remains constant. The graph concludes that under these conditions, the deduplication process is extremely fast and showcases the speed and efficiency of the algorithm when duplicates are most.

The execution time for the average case in Figure 6 scenario depicts a realistic measurement of the deduplication system's performance under conditions that are ordinary. Here, some of the input query videos are redundant copies, while the rest are unique. This represents a common use case. It shows the system's efficiency in handling a moderate amount of duplicate video content, concluding that the system maintains a balanced performance by processing duplicates efficiently without notable delays. The initial videos are found to take a considerable execution time, but this improves for the subsequent videos, by only requiring a minimal execution time which is almost constant.

Figure 7 highlights the system's performance under conditions with no duplicate videos at all. It presents the most difficult situation for the algorithm, where the system has to store all of the input query videos to the database, as each of these videos are unique to each other. The chart concludes that while the execution time required in this case is increased on a whole, the system remains robust, effectively managing such datasets without significant performance degradation. It is found to take higher execution time for the initial query videos, but there is notable reduction in the time required, as the following videos are processed.

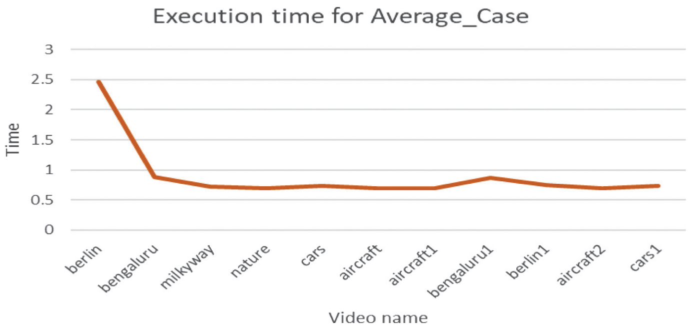

Fig. 6. Execution time for Average_Case.

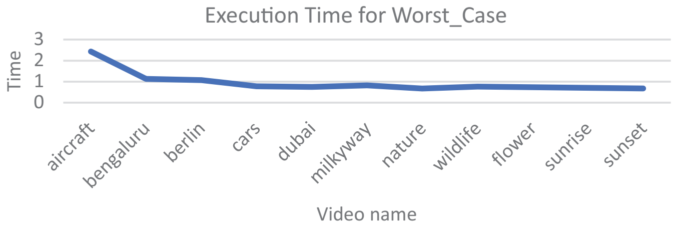

Fig. 7. Execution time for Worst Case.

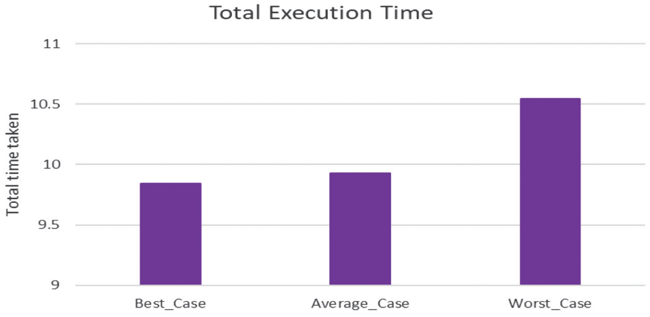

Fig. 8. Total execution time.

Three distinct scenarios are compared in Figure 8 "Total Execution Time": Best Case, Average Case, and Worst Case. With an execution time of just over 9.5 seconds, the Best-Case scenario has the shortest time. It takes longer, about 10 seconds, in the Average-Case scenario. With an execution time of almost 10.5 seconds, the Worst-Case scenario has the longest. The system operates most effectively in the Best-Case scenario, according to this comparison, with performance gradually declining in the Average- and Worst-Case situations.

The execution timings for several videos under three distinct conditions are displayed in Figure 9 named "Comparison of Execution Time for Best Case, Average Case, and Worst Case." While other videos have lower, but varying, execution times, the Worst Case (in purple) has much greater times for the "aircraft" and "Beijing" films, with times as high as 2.5 seconds. The "Beijing" video's Average Case (highlighted in red) has the longest execution duration, peaking at 2.5 seconds. The other movies have intermediate execution times. With very little differences, the Best Case (shown in blue) exhibits short execution times in every video. With the Best-Case scenario showing the fastest processing times across all movies, this graph illustrates the performance variations between the scenarios. In Fig. 11, the deduplication efficacy is compared for the Best-Case, Average-Case, and Worst-Case situations.

The deduplication ratio is a measure of the effectiveness of a deduplication process. It compares the amount of data before deduplication to the amount of data after deduplication. The formula for the deduplication ratio is shown in Equation 3:

$$\textit{Deduplication ratio} = \frac{\text{Total Size of All Data Before Deduplication}}{\text{Total Size of Unique Data After Deduplication}} \quad (3)$$

This ratio indicates how much storage space is saved through the deduplication process. A higher deduplication ratio signifies greater space savings. For example, a deduplication ratio of 4:1 means that the deduplication process has reduced the data size by a factor of 4.

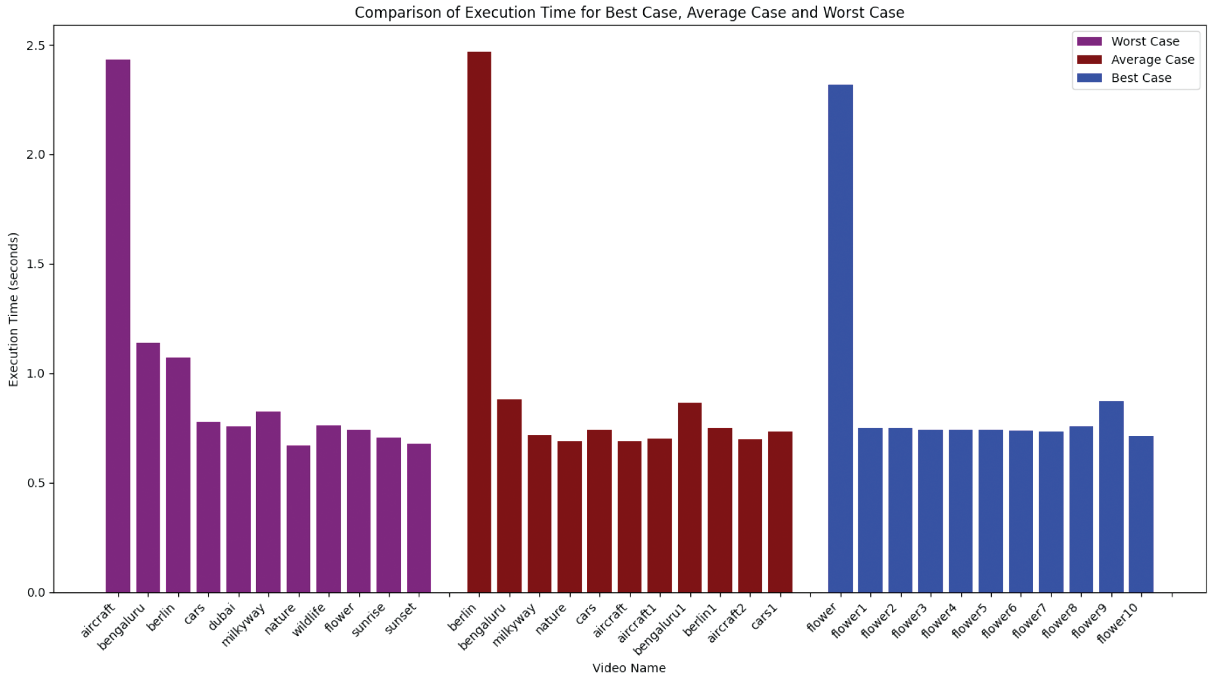

Fig. 9. Comparison of execution time.

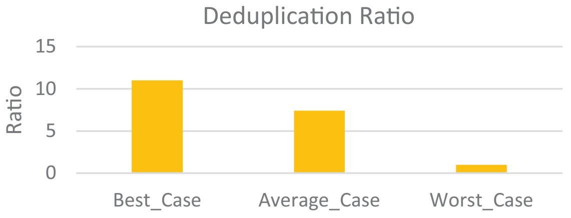

Fig. 10. Deduplication ratio.

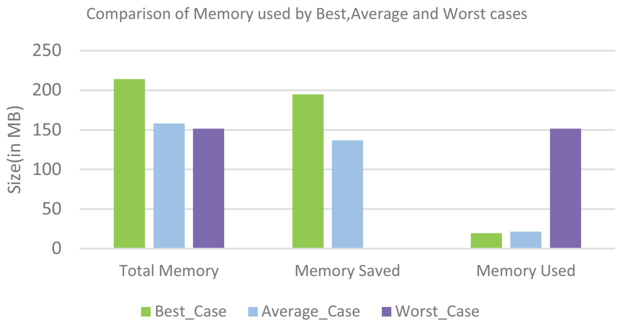

Fig. 11. Comparison of memory usage.

With a deduplication ratio of 11:1, the Best-Case scenario has the greatest, suggesting a notable decrease in redundant data. A lower deduplication ratio, of 7:1, is seen in the Average-Case scenario, indicating moderate efficacy. The deduplication ratio in the worst-case scenario is the lowest – it is 1:1, suggesting very little deduplication. With the Best Case getting the largest data reduction, the Average Case coming in second, and the Worst Case doing the least well, this comparison demonstrates the uneven effectiveness of the deduplication process.

The memory utilization in the best, average, and worst-case scenarios is contrasted in Figure 12. Memory Used, Memory Saved, and Total Memory are displayed on the x-axis. Size is shown on the y-axis in megabytes. The best case utilizes the most Total Memory (215 MB), followed by the worst case (150 MB) and average case (160 MB). Memory Saved shows that the worst scenario has no data shown, the average case saves roughly 140 MB, and the best case wins again with 195 MB. Remarkably, the pattern is reversed when it comes to Memory Used: the worst scenario requires 150 MB of memory, while the best and medium cases use much less (approximately 20–25 MB each). This implies that even though the optimal situation has the largest memory allocation overall, it saves memory more effectively, resulting in the lowest memory consumption.

A detailed examination of performance metrics for a system handling different video inputs is provided by the set of graphs. Because SHA-256 uses less memory than the current model, it is more efficient at storing hash values. The average and worst-case execution times exhibit a similar pattern: the first processed video (identified as “aircraft” in the worst case and “Berlin” in the average case) has an initial spike, which is followed by stabilized, reduced execution times for the others. This points to a system component that may be learning or adaptable. An interesting dynamic can be seen in the memory usage comparison: even though the best-case scenario has the largest overall memory allocation, effective memory-saving strategies actually cause it to consume the least amount of memory.

On the other hand, even though it has the lowest total allocation, the worst-case scenario utilizes the most RAM. Furthermore, the deduplication ratio study demonstrates that the best-case scenario reduces data the most, demonstrating a notable level of efficiency in reducing redundant data. Overall, these results point to a system that, while it may not be as fast to execute code, might still have memory management issues in extreme circumstances. The present research yields significant insights for prospective optimization tactics, specifically with regard to enhancing memory use for worst-case scenarios and preserving the efficiency attained upon processing initial inputs.

V. CONCLUSION

Video deduplication is an important feature of current cloud storage management, as it improves storage economy, reduces redundancy, and increases system scalability. As the volume of multimedia data grows exponentially, effective deduplication methods become increasingly more important. By employing advanced approaches such as meta-parameter clustering and GOP-level hashing, the suggested strategy improved storage optimization while maintaining efficient video content retrieval. However, some issues persisted, notably when seemingly comparable videos with different information were disregarded. The proposed methodology was designed to identify only complete video duplicates, meaning videos that were exactly identical in content, length, and encoding rather than near-duplicates, which may have minor variations in encoding settings, resolution, or metadata. Here, encoding referred to the method used to compress and store a video file using specific parameters such as codec, bitrate, resolution, and compression settings, which affected file size, quality, and playback characteristics. Since the clustering procedure is based on meta-parameters such as video length and frames per second (fps), videos with almost identical content but with minor differences in encoding settings or metadata may not be identified as duplicates, potentially leading to storage inefficiencies.

Moving forward, resolving these difficulties would necessitate continuous improvement of deduplication algorithms to improve precision and adaptability. Future research should look on hybrid systems that combine content-aware analysis with classic hashing techniques, allowing for more intelligent and accurate duplicate detection. Incorporating artificial intelligence and deep learning models could improve the deduplication process by allowing the system to discover commonalities other than metadata-based clustering, resulting in more extensive redundancy reduction. Furthermore, perfecting deduplication algorithms for large-scale distributed storage settings, such as cloud-based video streaming services and surveillance archives, will be critical to ensuring cost-effectiveness and performance. Deduplication systems may stay robust, efficient, and scalable by constantly improving and adopting new technologies, ensuring that cloud storage solutions match the expanding demands of modern data-driven applications.

CONFLICT OF INTEREST STATEMENT

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

REFERENCES

[1] G. Sujatha et al., “An efficient cloud storage model for GOP-Level video deduplication using adaptive GOP structure,” *Cybern. Syst.*, vol. 54, pp. 1–26, 2023.

[2] L. Chen, F. Xiang, and Z. Sun, “Image deduplication based on hashing and clustering in cloud storage,” *KSII Trans. Internet Inf. Syst.*, vol. 15, no. 4, pp. 1448–1463, 2021.

[3] J. K. Periasamy, and B. Latha, “Efficient hash function–based duplication detection algorithm for data deduplication deduction and reduction,” *Concurr. Comput.: Pract. Exp.*, vol. 33, no. 3, pp. e5213, 2021.

[4] S. Hema, and A. Kangaiammal, “Distributed storage hash algorithm (DSHA) for file-based deduplication in cloud computing,” in *Second International Conference on Computer Networks and Communication Technologies: ICCNCT 2019*, Coimbatore, India: Springer International Publishing, 2020, pp. 572–581.

[5] W. Xia et al., “The design of fast content-defined chunking for data deduplication-based storage systems,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 9, pp. 2017–2031, 2020.

[6] Y. Tan et al., “Improving the performance of deduplication-based storage cache via content-driven cache management methods,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 1, pp. 214–228, 2020.

[7] Z. Ke, H. Gong, and D. H. Du, PM-Dedup: Secure Deduplication with Partial Migration from Cloud to Edge Servers. arXiv preprint arXiv:2501.02350, 2025.

[8] J. Wu, and Y. Fu, “SGX based cloud-edge collaborative secure deduplication,” in *Workshop Proceedings of the 53rd International Conference on Parallel Processing*, Gotland Sweden: Association for Computing Machinery New York NY United States, 2024, pp. 112–113.

[9] Z. Yang, J. Li, and P. P. Lee, “Secure and lightweight deduplicated storage via shielded {deduplication-before-encryption},” in *2022 USENIX Annual Technical Conference (USENIX ATC 22)*, CARLSBAD, CA: USENIX – the Advanced Computing Systems Association, 2022, pp. 37–52.

[10] J. Wang et al., Survey of Disaggregated Memory: Cross-layer Technique Insights for Next-Generation Datacenters. *arXiv preprint arXiv:2503.20275,* 2025.

[11] R. Aparna, S. Bandopadhyay, and S. Pandey, “BlockDrive: A deduplication framework for Cloud using edge-level blockchain,” in *2021 International Conference on Communication Information and Computing Technology (ICCICT)*, IEEE, 2021, pp. 1–6.

[12] Z. Tang et al., “Fuzzy deduplication scheme supporting pre-verification of label consistency,” in *International Conference on Provable Security*, Cham: Springer Nature Switzerland, 2023, pp. 365–384.

[13] M. Allouche, and M. Mitrea, “Video fingerprinting: Past, present, and future”, *Front. Signal Process.*, vol. 2, pp. 984169, 2022.

[14] H. Shin, D. Koo, and J. Hur, “Secure and efficient hybrid data deduplication in edge computing”, *ACM Trans. Internet Technol. (TOIT).*, vol. 22, no. 3, pp. 1–25, 2022.

[15] A. Qin et al., “Maze: A cost-efficient video deduplication system at web-scale,” in *Proceedings of the 30th ACM International Conference on Multimedia*, Lisboa Portugal: Association for Computing Machinery New York NY United States, 2022, pp. 3163–3172.

[16] G. Sujatha, and J. R. Raj, “Challenges in implementing video deduplication in cloud storage system”, *Int. J. Syst. Syst. Engn.*, vol. 11, no. 3–4, pp. 399–416, 2021.

[17] P. J. Sriraksha, S. Chaudhari, and R. Aparna, “Video Deduplication using CNN (Conv2d) and SHA-256 hashing,” in *2022 Fourth International Conference on Cognitive Computing and Information Processing (CCIP)*, IEEE, 2022, pp. 1–8.

[18] Q. Huang, Z. Zhang, and Y. Yang, “Privacy-preserving media sharing with scalable access control and secure deduplication in mobile cloud computing,” *IEEE Trans. Mob. Comput.*, vol. 20, no. 5, pp. 1951–1964, 2020.

[19] S. Long et al., "A similarity clustering-based deduplication strategy in cloud storage systems," in *2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS)*, Hong Kong: IEEE, 2020, pp. 35–43.

[20] J. Malhotra, and J. Bakal, "A survey and comparative study of data deduplication techniques," in *2015 International Conference on Pervasive Computing (ICPC)*, IEEE, 2015, pp. 1–5.

[21] S. Singh, and J. C. Kavitha, "Rapid video deduplication based on global & local features using convolution neural network", in *2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN)*, Vellore, India: IEEE, 2023, pp. 1–7.

[22] R. Guan et al., Contrastive multi-view subspace clustering of hyperspectral images based on graph convolutional networks. IEEE Transactions on Geoscience and Remote Sensing. 2024.

[23] Z. Luo et al., "Soc: Semantic-assisted object cluster for referring video object segmentation", *Advances Neural Inf. Process. Syst.*, vol. 36, pp. 26425–26437, 2023.

[24] Z. K. Eisham et al., "Chimp optimization algorithm in multilevel image thresholding and image clustering", *Evolving Syst.*, vol. 14, no. 4, pp. 605–648, 2023.

[25] J. Brieva, "Datamining and its applications", *J. Artificial Intelligence Technol.*, vol. 2, no. 3, pp. 77–79, 2022.

[26] D. Vijayakumar, K. G. Srinivasagan, and K. Vivekrabinson, "Enhancing cloud storage security through blockchain-enabled data deduplication and auditing with a fair payment", *Peer-to-Peer Netw Appl.*, vol. 18, no. 3, pp. 147, 2025.

[27] Q. Zhang et al., Blockchain-based Privacy-preserving Deduplication and Integrity Auditing in Cloud Storage. *IEEE Transactions on Computers,* 2025.

[28] B. Liu et al., Blockchain-Assisted Fine-Grained Deduplication and Integrity Auditing for Outsourced Large-Scale Data in Cloud Storage. IEEE Internet Things Journal, 2025.

[29] J. Lapmoon, and S. Fugkeaw, A Verifiable and Secure Industrial IoT Data Deduplication Scheme with Real-time Data Integrity Checking in Fog-Assisted Cloud Environments. *IEEE Access,* 2025.

[30] M. Song et al., SimLESS: a secure deduplication system over similar data in cloud media sharing. *IEEE Transactions on Information Forensics and Security,* 2024.

[31] X. Zheng et al., DIADD: Secure Deduplication and Efficient Data Integrity Auditing with Data Dynamics for Cloud Storage. *IEEE Transactions on Network and Service Management,* 2025.

[32] S. Ruba, and A. M. Kalpana, "Advanced chunk-based data deduplication framework for secure data storage in cloud using hybrid heuristic assisted optimal key-based encryption", *Wirel. Netw.*, vol. 31, no. 4, pp. 1–23, 2025.

[33] X. Tang et al., "A secure and lightweight cloud data deduplication scheme with efficient access control and key management", *Comput. Commun.*, vol. 222, pp. 209–219, 2024.

[34] M. H. Mohiuddin, and L. Tamilselvan, "IDedupNet: A mobilenetV3-based deep learning framework for efficient image deduplication in cloud computing environments", *Informatica*, vol. 49, no. 13, pp. 143–162, 2025.

[35] R. Kaur, J. Bhattacharya, and I. Chana, " Deep CNN based online image deduplication technique for cloud storage system", *Multimedia Tools Appl.*, vol. 81, no. 28, pp. 40793–40826, 2022.

[36] S. Chaudhari, and R. Aparna, "Survey of image deduplication for cloud storage", *Syst. Res information technologies*, no. 4, pp. 113–134, 2023.