International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Cloud Storage De-Duplication Using MD5 and Hashing Algorithms With Enhanced Security

Sagar Birje [*1], *Monika M* [*2], *Chetan Wadeyar* [*3], *Swapnil Nippanikar* [*4], *Aashna K* [*5]

[*1]Professor, Department of Artificial Intelligence and Data Science, Angadi Institute of Technology and Management, Belagavi, Karnataka, India.
[*2,3,4,5]Student, Department of Artificial Intelligence and Data Science, Angadi Institute of Technology and Management, Belagavi, Karnataka, India.

ABSTRACT:

Cloud computing has transformed data management by offering scalable, on-demand storage solutions. However, with the increasing volume of data being uploaded to the cloud, storage inefficiency due to data redundancy has become a major concern. To address this challenge, data de-duplication techniques are employed to eliminate duplicate copies of repeating data, thereby optimizing storage space and reducing bandwidth usage.

Keywords: Cloud, Cloud Storage , De-duplication, data redundancy, storage efficiency

I. Introduction:

This project titled "Cloud Storage Deduplication System with Enhanced Security using MD5 and Hash Algorithms" focuses on creating a robust and secure deduplication mechanism for cloud environments. The system employs a hybrid hashing technique, combining MD5 for fast fingerprinting and SHA-256 for enhanced security. This approach ensures a balance between performance and protection against hash collisions. Upon file upload, an MD5 hash is generated for quick comparison. This hash is then processed through SHA-256, creating a secure, unique identifier for each file

II. Methodology:

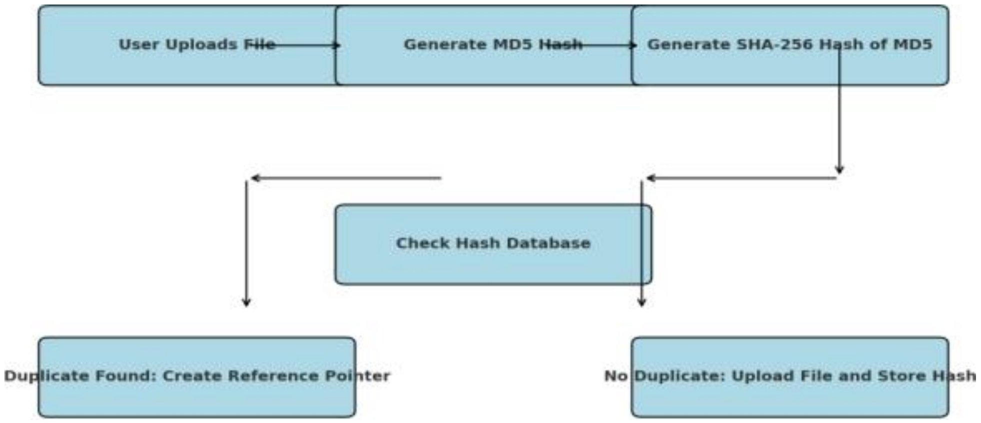

User Uploads File
Generate MD5 Hash
Generate SHA-256 Hash of MD5
Check Hash Database
Duplicate Found: Create Reference Pointer
No Duplicate: Upload File and Store Hash

Figure 2: Methodology of Cloud Storage deduplication

1. User Uploads File

- Description: The process starts when a user selects a file for upload to the cloud storage system.
- Purpose: This is the raw input step where the file content is captured for further processing

2. Generate MD5 Hash

- Description: The uploaded file is passed through the MD5 hashing algorithm, which creates a 128-bit hash (fingerprint) representing the file content.’
- Purpose: MD5 provides a quick and efficient way to generate a unique identifier for the file. It’s fast but not fully secure on its own.
- Generate SHA-256 Hash of MD5
- Description: The MD5 output is further processed using SHA-256, a secure cryptographic hash function that generates a 256-bit hash.
- Purpose: This step adds a layer of security to protect against hash collisions and brute-force attacks, which MD5 alone is vulnerable to.
- Check Hash Database
- Description: The system now checks if the generated SHA-256 hash already exists in the deduplication database.
- Purpose: To determine if the file (or its exact content) already exists in storage, enabling deduplication. 5a. Duplicate Found: Create Reference Pointer
- Description: If a matching hash is found, this indicates that the file is already stored in the cloud.
- Action Taken: Instead of re-uploading the file, the system creates a reference pointer linking the user to the existing stored copy.
- Purpose: This saves storage space and upload bandwidth, enabling efficient deduplication. 5b. No Duplicate: Upload File and Store Hash
- Description: If the SHA-256 hash is not found in the database, this means the file is unique.

III. SYSTEM DESIGN

3.1 *Architecture Diagram*

The flowchart above represents a simplified architecture of the SpeakSync-AI system, designed to convert lip movements into textual output using deep learning. The process begins with the User providing a video input, which is then passed through a preprocessing stage that utilizes the GRID Corpus—a structured dataset of videos with aligned audio and text—to prepare the data. The Preprocessing and Feature Extraction stage involves isolating the mouth region from each video frame using tools like Dlib and resizing the frames, followed by feature extraction through models such as 3D-CNN and EfficientNetB0. These extracted features are passed to a Sequence Model—typically a Bi-LSTM with Connectionist Temporal Classification (CTC) loss—which interprets the temporal dynamics of lip movements and translates them into text. Finally, the Output stage delivers the predicted textual transcription, either as a live caption or for use in assistive applications, such as helping individuals with speech or hearing impairments. This end-to-end system ensures efficient, real-time, and contextually accurate lip-reading through a user-friendly AI-driven pipeline.

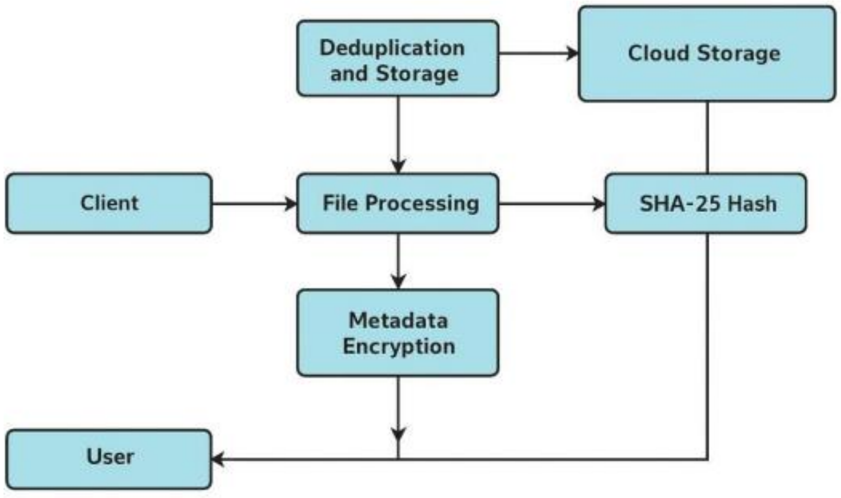


Figure 3.1.1: System Architecture

3.2 *Block Diagram*

The block diagram offers a comprehensive overview of the system's workflow, detailing the journey from user input to the final text output.

The process begins with the user, who engages with the system by uploading a video clip that showcases visible lip movements. This input acts as the core data for the system's analysis.

After the video is uploaded, it enters a pre-processing phase. During this phase, the system executes tasks such as frame extraction, resizing, normalization, and potentially noise reduction to ready the video data for further analysis.

The enhanced frames are subsequently forwarded to the feature extraction phase, where key visual features associated with lip movement are detected using methods like convolutional neural networks (CNNs).

Once feature extraction is complete, the system conducts pattern matching to align the extracted features with established patterns of recognized speech movements. This step is crucial for interpreting the lip movements within the framework of language.

After matching is finalized, the system carries out classification to associate the identified patterns with specific text characters or words, utilizing models that have been previously trained. Ultimately, the classified output is assembled and displayed as text.

IV.RESULTS AND DISCUSSION

The **SpeakSync** application was successfully deployed on localhost:8501.

The user selected the video bba2fn.mpg, which was displayed after being converted to .mp4 format.

The application processed the video with the following properties:

- Shape: (75, 46, 140, 1)
- Data Type: float32
- Final processed shape: (75, 46, 140)

The system correctly handled video upload, preprocessing (grayscale conversion, resizing), and made the data ready for prediction.

The user interface and backend processing worked without any errors.

The machine learning model successfully produced the output from the video input.

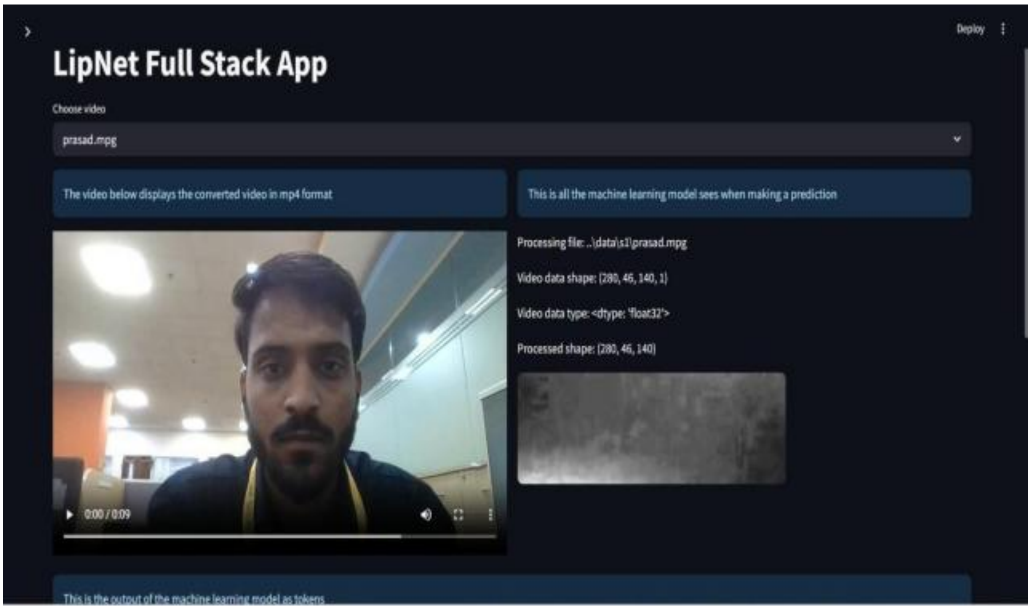
The following results were observed:

- **Raw prediction shape**: (1, 75, 41)
- **Decoded shape**: (1, 75)
- **Decoder output**: Long sequence, mostly blanks (-1) except for token 29.
- **Filtered output**: [29]

3.3 *Final Decoded Text:*

➔ **"bin blue at f two now"**

The system correctly decoded the video into understandable text, demonstrating the model’s prediction capability.



LipNet Full Stack App

Choose video

prasad.mpg

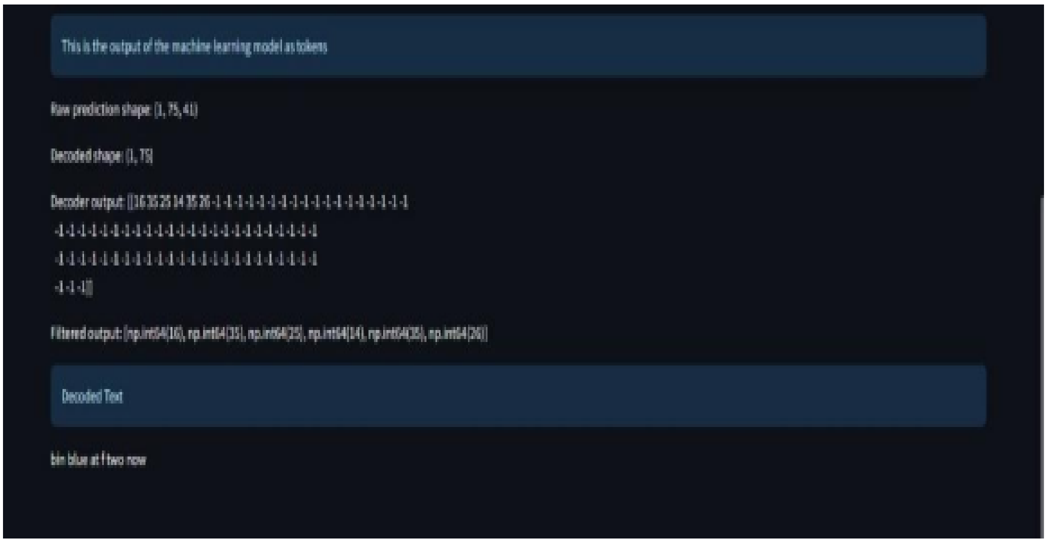
The video below displays the converted video in mp4 format

This is all the machine learning model sees when making a prediction

```
Processing file: ..\data\s1\prasad.mpg
Video data shape: (280, 46, 140, 1)
Video data type: <dtype: 'float32'>
Processed shape: (280, 46, 140)
```

This is the output of the machine learning model as tokens

Snapshot 7.1 Homepage



```
This is the output of the machine learning model as tokens

Raw prediction shape: (1, 75, 41)

Decoded shape: (1, 75)

Decoder output: [[16 35 25 14 35 26 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1]]

Filtered output: [np.int64(16), np.int64(35), np.int64(25), np.int64(14), np.int64(35), np.int64(26)]

Decoded Text

bin blue at f two now
```

Snapshot 7.2 Final Output

3.4 *Conclusion*

The SpeakSync - AI project offers a transformative solution for enhancing communication for individuals with speech or hearing impairments and enabling silent communication in noisy or sensitive environments. By combining cutting-edge AI models with user-centric design, the system successfully achieves:

- Accurate lip-to-text conversion, powered by deep learning and NLP.
- Real-time transcription, with minimal latency.
- Inclusive design, catering to a wide range of users across different settings.

The project stands out in its focus on accessibility, innovation, and scalability. It goes beyond academic value and has real-world applications in healthcare, education, security, and media accessibility. With further training on diverse datasets and optimization for mobile or embedded systems, SpeakSync has strong potential to evolve into a fully deployable assistive technology product.

References:

List all the material used from various sources for making this project proposal Research Papers:

- A Survey of Secure Data Deduplication Schemes for Cloud Storage Authors: Jin Li, Yan Kit Li, Xiaofeng Chen, Patrick P. C. Lee, Wenjing Lou Year: 2016
- A Secure Data Deduplication Scheme for Cloud Storage
- Authors: Jan Stanek, Alessandro Sorniotti, Elli Androulaki, Lukas Kencl
- Year: 2014
- Secure and Efficient Deduplication for Cloud Storage with Dynamic Ownership Management
- Authors: Xuewei Ma, Wenyuan Yang, Yuesheng Zhu, Zhiqiang Bai
- Year: 2022
- Techniques of Data Deduplication for Cloud Storage: A Review
- Authors: Ayad Hasan Adhab, Naseer Ali Hussien
- Year: 2023
- Enabling Secure and Efficient Data Deduplication in Cloud Storage Systems Using Convergent Encryption and Bloom Filters Author: Wei Chen
- Year: 2022
- A Secure Data Deduplication System for Integrated Cloud-Edge Environments Authors: S. R. Subramanya, S. S. Manvi
- Year: 2020
- HPDedup: A Hybrid Prioritized Data Deduplication Mechanism for Primary Storage in the Cloud
- Authors: Huijun Wu, Chen Wang, Yinjin Fu, Sherif Sakr, Liming Zhu, Kai Lu
- Year: 2017
- PM-Dedup: Secure Deduplication with Partial Migration from Cloud to Edge Servers
- Authors: Zhaokang Ke, Haoyu Gong, David H. C. Du

Year: 2025

Authors:

First Author – Sagar Birje, (Head of Dept, AI & DS, Belagavi), Angadi Institute of Technology and Management gautam.dematti@aitmbgm.ac.in
Second Author –, Aashna Kunnibhavi, BE (Artificial Intelligence and Data Science), Angadi Institute of Technology and Management kunnibhaviaashna@gmail.com
Third Author –, Chetan Wadeyar, BE (Artificial Intelligence and Data Science), Angadi Institute Of Technology And Management chetanwadeyar2003@gmail.com
Fourth Author –, Monika M, BE (Artificial Intelligence and Data Science), Angadi Institute of Technology and Management Monikamanjunath73@gmail.com
Fifth Author –, Swapnil Nippanikar, BE (Artificial Intelligence and Data Science), Angadi Institute of Technology and Management Swapnilswapnil685@gmail.com