Echo Halt: A Method for Reimagining Social Media Video Content Without Duplication

Debayan Mukherjee, Dhanushvarma Ravivarma and Isaac Gonzalez* , Arizona State University, USA

"Duplication is not being Inspirational" - Team Echo Halt.

Video continues to dominate the social media content landscape, especially short-form video. The explosive nature of video as a content on these platforms have created a rather massive challenge to moderate, enforce copyright and digital forensics. One issue that has emerged is the prevalence of content duplication. Content creators find new and innovative ways to create engaging video content with elaborative to small scale production requirements for their viewers, followers and target audience while aiming to gain the maximum engagement and web traffic. Same content or near same content uploaded by multiple users or accounts, often without proper recognition has been creating a diversion of engagement, source identification and most importantly recognition. Social media platforms have over the years engaged in duplicate or near-duplicate video identification (NDVI), however the platforms generally do not penalize or restrict such users or content. This paper presents an approach that identifies video duplicates or overlaps in short form video content before they can 'go-live' on the social media platform. The program sets up a multi-tiered approach using deep learning, computer vision techniques and machine learning models. Our method includes the development of a robust feature extraction framework that can handle common video transformations, such as cropping, resizing, and minor frame manipulations, which are often used to bypass existing duplicate content detection methods. We evaluate the effectiveness of our approach against a repository of videos from a popular platform (Instagram, Reels), demonstrating its accuracy and efficiency in identifying duplicates. The results provide an intuitive user experience for content creators who can analyze the overlap quotient of their upload against the platform's current repository. The platform also provides pre-production tips and trending video analysis to users such that NDVI is reduced at that stage. This research has implications for improving content curation, intellectual property protection, and the overall user experience on social media platforms.

CCS CONCEPTS • Computing Methodologies • Visual Content-based indexing and retrieval • Human-centered computing • Social Media • Content analysis and feature selection • Information systems.

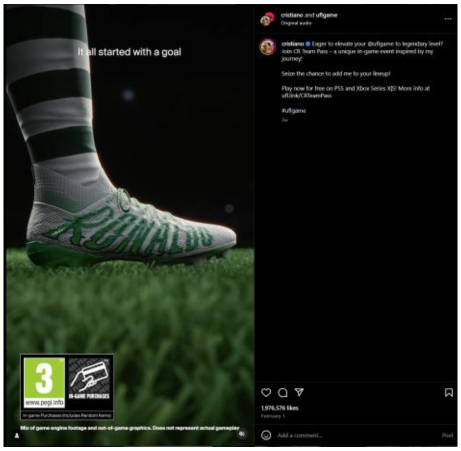
ACM Reference Format:

Debayan Mukherjee, Dhanushvarma Ravivarma and Isaac Gonzalez. Echo Halt: Method and reimagining social media video content without duplication.

1.1 Introduction

Anyone with a social media profile on Instagram, TikTok, YouTube and Snapchat who has scrolled through the feed or the 'for you' sections know how big a role video plays in social media. In the last decade, short form video has taken over the content landscape. As the number of online bloggers and organizations increases, so does the volume of online video content. For instance, videos could generate even one billion views in "TikTok" daily [1]. TikTok, one of the most

Authors' Contact Information: Debayan Mukherjee, dmukhe16@asu.edu; Dhanushvarma Ravivarma, dravivar@asu.edu; Isaac Gonzalez, abc@asu.edu; Arizona State University, Tempe, Arizona, USA.



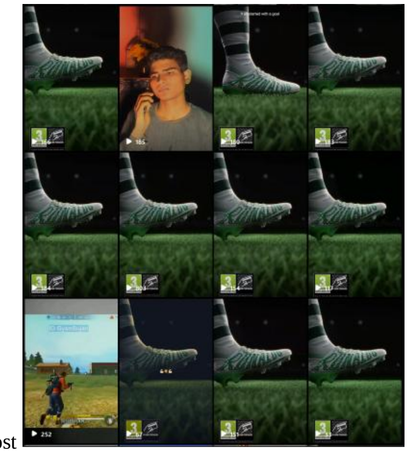
Original Post

Reposts

Fig 1. The left image shows the original post on the creator's page with 1.9MM likes and 56MM plays and there are a minimum ~26 reposts or altered video versions of the video with the same audio with maximum ~1K views. However, all the reposted videos have received some view engagements.

important and representative short-form video platforms across the globe, has acquired a user base of more than 1.5 billion by 2023 with a 16% year-wise growth [2]. Deep learning has pushed the design of new methods that can learn forensic fingerprints automatically from data [3], [4] helping us to take a new step towards applying these techniques to real problems. Despite the promising results of artificial neural networks, some limitations remain. Recently, collecting data from social networks has become increasingly difficult because of data protection regulations and the most stringent policies introduced by the platforms [5]. Indeed, it is mandatory to obtain end user consent or the platform's written permission before acquiring data via the API or web scraping of the most common social networks like Facebook, Instagram or Twitter.

500 hours of video are uploaded to YouTube every minute [6], TikTok users consume approximately 167 million hours of content daily [7] and one billion hours of content was watched on YouTube every day in February 2017. With billions of videos being available on the internet, it becomes a major challenge to perform NDVI. In recent years, short videos with durations of less than 60 seconds have become increasingly popular on social media platforms. Users can easily copy a hot short video and upload a modified version on these platforms to gain attention as shown in Fig. 1. With the increasing in short video data, there appear new difficulties and challenges for detecting near-duplicate short videos. Some of the new difficulties and challenges are listed as follows. Firstly, most long videos are generated by professional videographers with elaborate production teams with cameras, while most short videos are generated by amateurs with mobile devices. Hence, the short videos might contain some new types of near duplicates, e.g., horizontal/vertical screen videos and camera shaking videos. Secondly, as the cost of editing a short video is cheaper, users might prefer to edit a short video. Hence, the number of near-duplicate short videos is larger than that of near-duplicate long videos.

To address these gaps, we reimagined a social media platform, Echo Halt. The platform accounts for NDVI and uses algorithms that addresses challenges in tackling NDVI. We employed a multi-algorithm approach combining hash computation and comparison resilient to basic transformations in the video content. Deep feature extractions, temporal analysis, keyframe analysis, manipulation aware comparisons and robust weighted decision system that enables users to critically evaluate their content's analysis on the platform. Unlike most single threaded approaches for NDVI, Echo Halt's multi-faceted strategy provides robust performance across multiple scenarios.

2 RELATED WORK

Social media companies employ a variety of advanced techniques to detect duplicate or near-duplicate video uploads. These solutions leverage machine learning, computer vision, and content-based algorithms to ensure the originality of uploaded content. Below is a detailed breakdown of the technical approaches: NDVR aims to retrieve the near-duplicate videos from a massive video database, where near-duplicate videos are defined as videos that are visually close to the original videos [8]. For example, the videos might be slightly modified by the users to bypass the detection, and the modified videos can be treated as near-duplicate videos of the original videos. These modifications can be inpainting (text) insertion, border insertion, masked human, and so on.

Social media platforms have implemented sophisticated systems to gather NDVI content, applying a multi-faceted approach that combines computational vision and machine learning techniques [9]. These systems typically employ content-based duplicate detection methods, including Convolutional Neural Networks (CNNs) for feature extraction that create distinctive video-level representations through aggregation techniques such as max-pooling [10]. The Bag-of-Visual-Words model generates histograms of visual features by creating visual codebooks through clustering algorithms, with cosine similarity used for comparison [11]. Perceptual hashing at the frame level creates transformation-invariant hashes to identify visually similar content despite minor modifications [12]. Platforms further enhance detection through metadata and encoding analysis, assigning unique Content IDs based on properties such as frame rate, pixel distribution, and audio waveform patterns[13]. These extracted features are efficiently stored in vector databases that utilize Approximate Nearest Neighbors (ANN) algorithms for similarity searches [14]. Audio fingerprinting techniques complement visual analysis by creating distinctive identifiers from audio waveforms, enabling detection even when visual elements have been altered [15]. These technological approaches are often supplemented by user reporting systems and manual review processes, creating a comprehensive framework for maintaining content originality and copyright compliance [16].

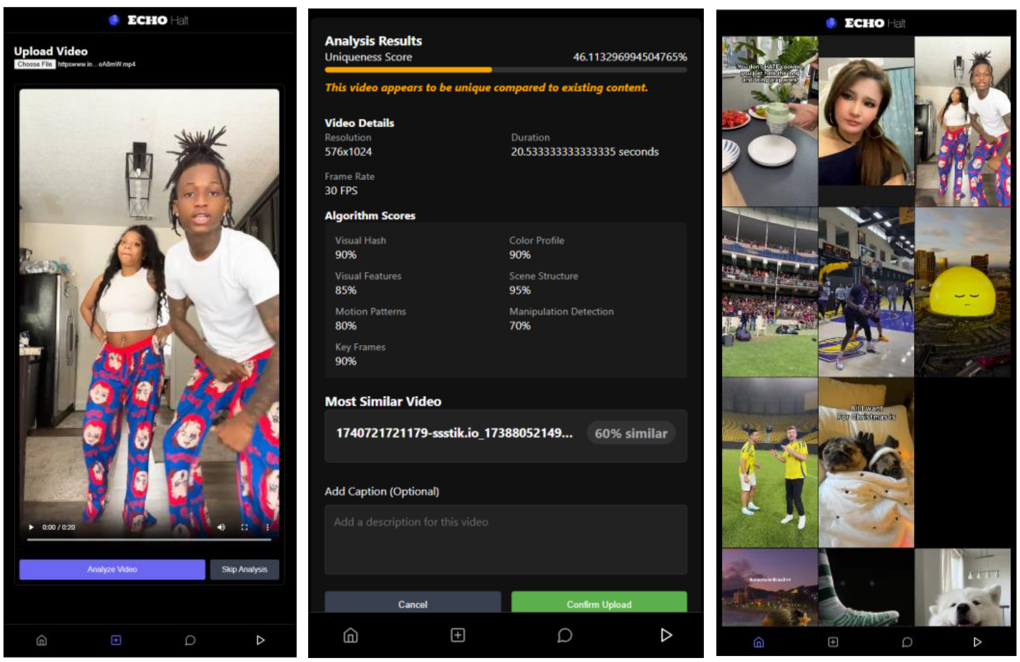
Despite considerable progress, existing approaches exhibit several critical limitations, most systems are not designed to detect specific manipulation strategies like split-screens, overlays, or strategic frame insertions. Many approaches provide binary "duplicate/non-duplicate" decisions without quantifying similarity levels or identifying manipulation types. Few systems effectively combine multiple algorithms to address diverse manipulation techniques. A lot of platforms operate as black boxes, providing little information about why videos are flagged as duplicates. Through our experiments, we aim to overcome these challenges.

3 OVERVIEW OF OUR PROCESS

Our video deduplication system implements a comprehensive approach that begins with an AI-assisted research phase, enabling content creators to review trending content and understand key elements before production. The technical foundation of our model combines four main layers, Frame extraction layer; Analysis layer; Manipulation detection; Decision layer. One of the most critical part of the system is to intelligently extract representative frames. The process begins with frame extraction and basic feature computation, followed by more complex similarity analyses, and culminates in specialized manipulation detection algorithms and an integrated decision-making process. The core insight of our approach is that no single algorithm can reliably detect all forms of video manipulation. By combining multiple complementary techniques—from perceptual hashing to deep feature extraction and specialized manipulation detectors, Echo Halt achieves robust performance across diverse manipulation scenarios.

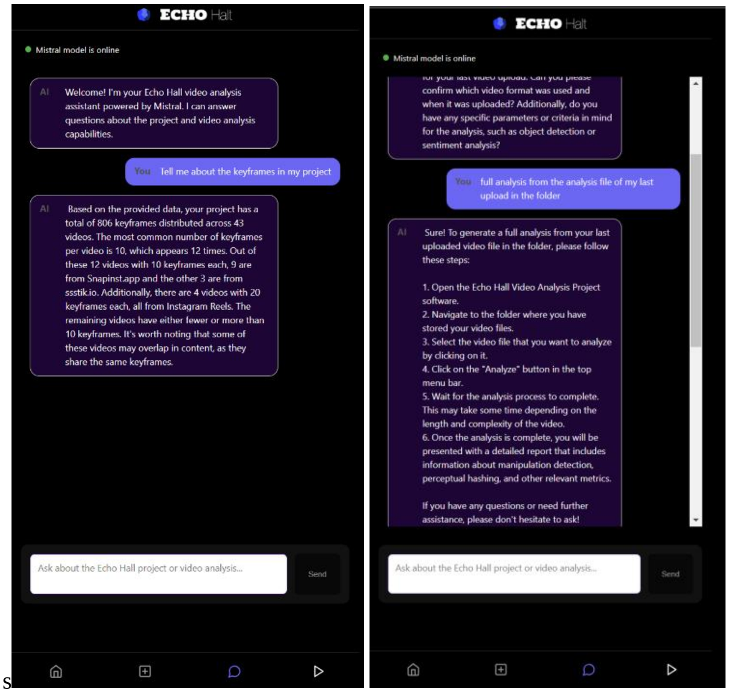
4 DEVELOPMENT OF THE METHOD

The front-end implementation of Echo Halt is designed to provide an intuitive and accessible user experience for duplicate content detection across social media platforms. Built using React.js with Redux for state management and Tailwind CSS for styling, the interface ensures a responsive and visually appealing design. Key design principles include mobile-first responsiveness, minimalist aesthetic. The system enables users to seamlessly upload videos, receive instant duplicate analysis feedback and manage content through an interactive dashboard (Fig 2). API communication is handled via Axios, ensuring smooth data retrieval, while WebSockets support real-time feedback. Visual indicators such as progress bars, color-coded uniqueness scores enhance usability. The front-end features home screen, which showcases the repository of unique videos in an animated thumbnail design. The click reactive thumbnail opens the video in a full screen format similar to popular social media platforms like Instagram and TikTok. There is also a chat navigation that connects the user to Mistral Instruct 7B AI chat agent. The chat agent is pre-connected to the project folder alone and responds only to project related questions as illustrated in Fig 3.



[User Uploads Video] → [Real-time Preview & Feedback] → [Duplicate Analysis] → [Visualization & Score] → [Upload to Video Repository]

Fig 2. Workflow for video uploads.



s

Fig 3. User interaction with Chat agent on project's keyframe details integrating Mistral Instruct 7B model.

4.1 FRAME EXTRACTION

Temporal-Focused Extraction - This method prioritizes frames from the beginning and end of videos, where manipulations commonly occur. The algorithm for temporal-focused extraction is shown below:

```
frame_count = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
fps = cap.get(cv2.CAP_PROP_FPS)
if focus_on_ends:
    start_count = int(max_frames * 0.4)  # 40% from beginning
    end_count = int(max_frames * 0.4)    # 40% from end
    middle_count = max_frames - start_count - end_count
    start_indices = [int((i / start_count) * (frame_count * 0.2))
                     for i in range(start_count)]
    end_indices = [int(frame_count * 0.8 + (i / end_count) * (frame_count * 0.2))
                   for i in range(end_count)]
```

Uniform Sampling - For videos with consistent content throughout, uniform sampling provides better coverage by extracting frames at regular intervals. This is particularly useful for establishing baseline similarity.

Scene-Based Extraction - For videos with sufficient duration, we use a scene change detection algorithm to identify keyframes at significant content transitions. This optimizes the representativeness of our frame sample while minimizing redundancy. Our adaptive approach selects the most appropriate extraction strategy based on video characteristics and available computational resources. For short-form social media videos (under 60 seconds), we typically employ a combination of temporal-focused and uniform sampling.

4.2 CORE ANALYSIS ALGORITHMS

Enhanced Perceptual Hashing - We implemented an improved perceptual hashing algorithm that expands upon traditional pHash by incorporating both horizontal and vertical gradients. Also incorporates a gradual similarity curve rather than simple Hamming distance, making it more sensitive to subtle differences while reducing the impact of isolated bit flips that often result from minor edits:

```
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
gray = cv2.GaussianBlur(gray, (3, 3), 0)
resized = cv2.resize(gray, (hash_size + 1, hash_size),
                    interpolation=cv2.INTER_AREA)
h_diff = resized[:, 1:] > resized[:, :-1]
resized_v = cv2.resize(gray, (hash_size, hash_size + 1),
                      interpolation=cv2.INTER_AREA)
v_diff = resized_v[1:, :] > resized_v[:-1, :]
combined_hash = np.concatenate([h_diff.flatten(), v_diff.flatten()])
```

Color Histogram Analysis –

```
hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
hist = cv2.calcHist([hsv], [0, 1, 2], None, [16, 16, 16],
                  [0, 180, 0, 256, 0, 256])
cv2.normalize(hist, hist, 0, 1.0, cv2.NORM_MINMAX)
```

```
    return hist.flatten()
  except Exception as e:
    print(f"Error computing histogram: {str(e)}")
    return np.zeros(16 * 16 * 16)
```

Deep Feature Extraction - We employ a pre-trained ResNet-18 model for extracting deep features from video frames. These features capture semantic content that remains consistent even when videos undergo significant visual transformations:

```
try:
   pil_image = Image.fromarray(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
         img_tensor = transform(pil_image).unsqueeze(0).to(device)
      with torch.no_grad():
          features = model(img_tensor)
```

Color Histogram Analysis - While less reliable as a standalone measure, color distributions provide valuable complementary information, particularly for distinguishing videos with similar spatial content but different color grading:

```
hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
hist = cv2.calcHist([hsv], [0, 1, 2], None, [16, 16, 16],
                  [0, 180, 0, 256, 0, 256]
cv2.normalize(hist, hist, 0, 1.0, cv2.NORM_MINMAX)
```

4.3 MANIPULATION DETECTION ALGORITHMS

Temporal Manipulation Detection - This component identifies videos that have been temporally manipulated through frame insertion, truncation, or reordering. It employs a combination of frame fingerprinting and Dynamic Time Warping (DTW) to detect temporal alterations even when they constitute only a small portion of the overall video.

Overlay Detection - The overlay detection algorithm identifies cases where text, logos, or graphics have been added to a video to obscure its similarity to the original.

Split-Screen Detection - This component identifies cases where multiple videos have been combined in a split-screen layout, a common technique for creating derivative content.

Reflection Detection - This algorithm identifies videos that have been horizontally flipped (mirrored), a simple but effective technique for evading basic detection systems

4.4 DECISION MAKING PROCESS

To determine a score for uniqueness benchmarked at 40% overall score we established:

Weighted Similarity Aggregation - We aggregate similarity scores from multiple algorithms using a weighted approach that prioritizes the most reliable signals

```
weights = {
        "perceptual_hash": 0.20, #More reliable for visual similarity
        "histogram": 0.05, #Less reliable on its own
        "deep_features": 0.30, #Most reliable for semantic content
        "scene_structure": 0.15, #Good for temporal structure
        "motion": 0.10, #Helps with dynamic content
        "keyframes": 0.15, # Good for key visual elements
        "manipulation": 0.05 # Accounts for detected manipulations
    }
```

Manipulation-Aware Decision Making - The final decision incorporates both similarity scores and detected manipulations, applying appropriate penalties based on the type and extent of manipulation:

```
base_penalty = 0
            if manip_type in ["frame_insertion", "temporal_truncation"]:
                # These significantly change content - higher penalty
                base_penalty += 3
            elif manip_type in ["overlay_modification", "reflection_transformation"]:
                # These modify presentation but keep core content - lower penalty
                base_penalty += 1.5
            else:
                # Other manipulation types - moderate penalty
                base_penalty += 2
        # Cap the total penalty
        manipulation_penalty = min(15, base_penalty)
        # Only apply penalty for videos with moderate to high similarity
        if overall_similarity > 60:
            # Apply full penalty for high similarity
            overall_similarity = max(0, overall_similarity - manipulation_penalty)
        elif overall_similarity > 40:
            # Apply reduced penalty for moderate similarity
            scaled_penalty = manipulation_penalty * ((overall_similarity - 40) / 20)
```

```
        overall_similarity = max(0, overall_similarity - scaled_penalty)
# Special case: Boost for highly similar deep features
deep_feature_sim = comparison_result.get("similarities", {}).get("deep_features", 0)
if deep_feature_sim > 75:
    deep_feature_boost = (deep_feature_sim - 75) / 25  # 0 to 1

    # The boost is stronger when perceptual hash also agrees
    phash_sim = comparison_result.get("similarities", {}).get("perceptual_hash", 0)
    if phash_sim > 70:
        hash_agreement = (phash_sim - 70) / 30  # 0 to 1
        boost_factor = deep_feature_boost * 0.3 * (1 + hash_agreement)

        # Apply the boost
        overall_similarity = min(100, overall_similarity * (1 + boost_factor))
# Determine duplicate status
is_duplicate = overall_similarity >= 65
uniqueness = 100 - overall_similarity
```

5 CREATING THE DATASET

Our experimental dataset for video deduplication testing was systematically developed using a diverse corpus of content spanning five distinct categories: technology products, beauty, pets, sports, and travel. The final collection comprised 16 unique video samples, with each original video having 2-3 deliberately manipulated duplicates existing on the Instagram platform. These variations were created through several controlled transformation techniques: temporal truncation that reduced video length while preserving key content; frame insertion at video beginnings to alter initial visual cues; overlay modifications through the addition of text and graphic elements; unaltered duplicates to establish baseline detection metrics; split-screen compositions that combined original content with other visual material; and reflection transformations that mirrored the visual content while maintaining semantic integrity. Audio track alterations were acknowledged but designated as beyond the current scope of analysis. This methodical approach to dataset creation enables comprehensive evaluation of our deduplication algorithms across a range of realistic manipulation scenarios commonly encountered in social media environments.

6 EVALUATING WITH REAL TIME USE CASE

The Echo-Halt implementation and evaluation proceeded through a structured three-phase approach to systematically isolate technical challenges and iteratively refine system components.

6.1 FRONTEND IMPLEMENTATION

The initial phase focused on establishing the React-based user interface and coordinating communication between multiple service endpoints. This phase encountered several technical challenges related to system coordination:

The frontend implementation required sophisticated state management for different viewing states. Developing tile-based interfaces with reactive controls presented significant challenges, particularly for the home and reels pages where content grid layouts needed to adapt dynamically to content characteristics while maintaining consistent interaction patterns. Our distributed architecture involved multiple services operating on distinct ports: React application (port 3003), Python video analysis server (port 3001), Node.js middleware (port 3002), and the AI agent service (port 3004). This configuration created complex cross-origin resource sharing (CORS) challenges. Implementing the upload functionality necessitated careful integration between frontend components and backend services. File uploads required large video analysis requirements to the Python analysis server while maintaining responsive user feedback and error handling.

6.2 VIDEO ANALYSIS ALGORITHM

The second phase focused on optimizing the video analysis pipeline and addressing performance limitations encountered during initial testing. Early implementations of our feature extraction algorithms exhibited inconsistent performance across different video uploads unique or non-unique. Histogram analysis and Perceptual hashing particularly struggled with videos featuring similar visual compositions but different content, generating excessive false positives.

Random analysis revealed a concerning rate of both false positives (unique content incorrectly identified as duplicates) and false negatives (actual duplicates not detected). Analysis determined that in many cases, the system was defaulting to fallback similarity scores when primary analysis components failed to complete within timeout thresholds. The intensive nature of deep feature extraction occasionally resulted in processing timeouts, particularly for higher-resolution videos. When these timeouts occur, the system would resort to less sophisticated fallback metrics, creating inconsistency in detection quality.

To address these challenges, we implemented several key refinements to the analysis pipeline. We modified service timeout parameters to dynamically adjust based on video characteristics (length, resolution, complexity), significantly reducing timeout-induced fallbacks. We implemented a more sophisticated debug system that preserved partial analysis results rather than defaulting to complete fallbacks when specific components failed. The similarity calculation algorithm was recalibrated to place greater emphasis on more reliable feature types when specific analyses exhibited high uncertainty or failed to complete.

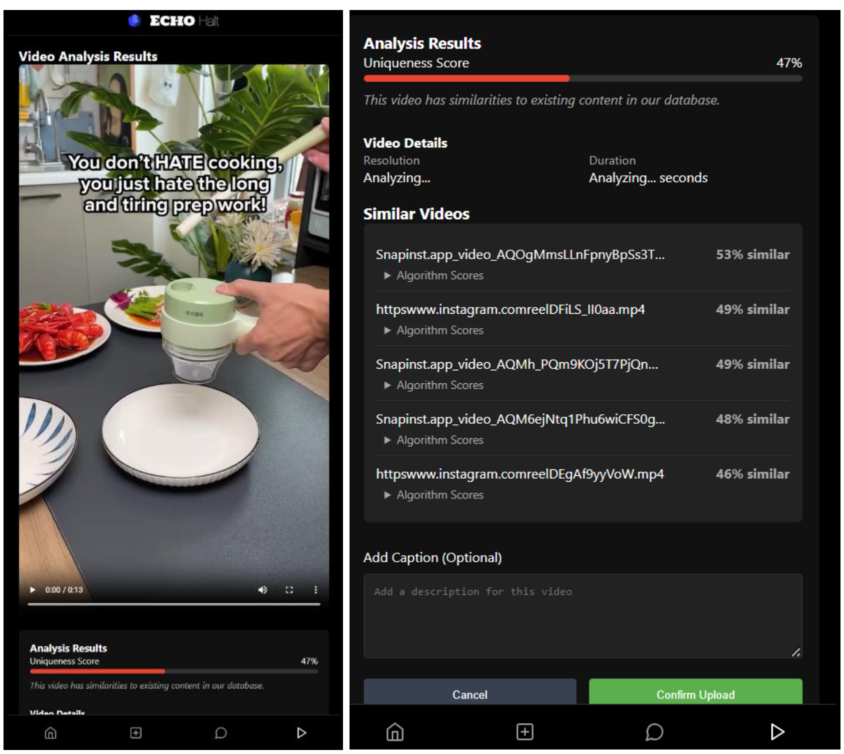


Fig 4: False positive with upload options.

6.3 AI AGENT INTEGRATION

The final phase focused on integrating conversational AI capabilities to enhance the user experience through natural language interaction and explanation. Initial attempts to integrate Llama 2 and Llama 3 models via Ollama encountered persistent connectivity issues, as 404 errors and unpredictable API behavior i.e. unresponsive instances. To ensure response effectiveness and relevance of model responses, we needed to implement context boundaries that confined the model's knowledge and references to specific project directories (video repository, analysis folders, and keyframe storage). Early integration tests revealed inconsistencies in response formatting and occasional hallucination of non-existent project components, creating potential confusion for users Fig 5.

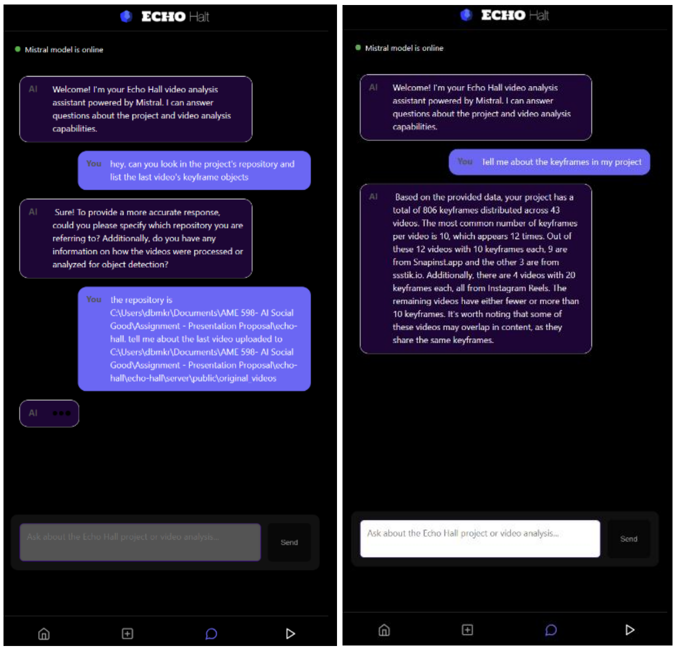


Fig 5: AI Agent response with undefined context (left). Defined Context (Right).

After evaluating several alternative approaches, we pivoted to implementing Mistral Instruct 7B using the GPT4All framework. The GPT4All integration provided more consistent API behavior and eliminated the 404 errors encountered with Ollama. We implemented a robust context injection system that provided the model with structured access to project directories and analysis outputs while preventing unauthorized access to system resources beyond specified boundaries.

7 LIMITATIONS AND FUTURE WORK

While Echo Halt establishes a prototype that analyses video content in a tiered manner, we believe a significant step towards establishing a platform could be achieved by developing further on this baseline. Despite our optimization efforts, Echo Halt's comprehensive approach requires more computational resources than simpler detection methods. Our current analysis requires approximately 80 seconds for a 720p video at 24 FPS. This establishes a pattern that prioritizes quicker analysis methods compared to further deep learning methods with a more elaborative algorithm. One optimization technique we used that helped speed the analysis process was to have repository video key frames (max. 10 frames per video) extracted and stored locally. While the prototype successfully detects many common manipulation techniques, it remains vulnerable to more innovative future techniques to evade algorithmic analysis. Our experiments with advanced adversarial manipulation combining multiple transformation types with parameters specifically tuned to minimize similarity, however for a full-scale establishment the program needs to continuously reiterate its approach. The prototype also ignores any audio signature analysis. We believe that user generated audio and music tracks in video are not primary checkpoints for a duplication callout. While copyright laws would establish audio usage further.

To further advance the efficiency and robustness of the Echo Halt system, one critical avenue involves neural network optimization, wherein techniques such as model development, quantization and knowledge distillation will be explored to advance analysis systems. Additionally, hardware acceleration will be leveraged to execute performance-critical operations with significantly lower latency. Furthermore, a streaming processing framework will be developed, enabling incremental analysis of video content during the upload process, thereby mitigating perceived latency and enhancing real-time responsiveness for end users. The system will also integrate detection mechanisms capable of recognizing deviations from expected patterns, thereby identifying sophisticated adversarial attempts that rely on subtle changes to the content.

Expanding upon the system's multimodal analysis capabilities, future iterations will incorporate audio fingerprinting to detect cross-modal duplications, ensuring that user-generated similarities are accounted for even in cases where visual content is altered. Additionally, text extraction techniques, including OCR-based caption analysis, will facilitate the identification of textual markers that suggest semantic relationships between different video instances. Lastly, a key research trajectory will focus on self-supervised learning as a means of enhancing feature extraction without reliance on extensive manually labeled datasets. Machine learning will be leveraged to refine the model's ability to distinguish between genuine duplicates and non-duplicates by training on structured positive and negative sample pairs. By integrating these advancements, Echo Halt will evolve into a more efficient in expanding NDVI across social media landscapes.

8 CONTRIBUTIONS

We established a forte based approach to tackle this project, Isaac headed the interface, experience and navigation flow designs along with Figma prototypes. He also headed the logo and other elemental designs of the interface and provided feedback during the Frontend buildout. He also took the lead in developing the final presentation for the project. Dhanush headed algorithm testing with repository videos as a first pass approach to arrive at the approach the analysis program will consider. He explored multiple tiered approaches to the solution as discussed in the paper and arrived at the final mechanisms for the analysis backend. He has also contributed to the technical documentation of this paper and presentation. Debayan headed the overall project management, frontend buildout, paper documentation, building the prototype package integrating backend, decision making logics and AI agent integration to the prototype.

9 CONCLUSIONS

This paper presented Echo Halt, a comprehensive system for NDVI and manipulated videos in social media environments. Traditional duplicate detection approaches that rely solely on visual similarity are insufficient for modern social media environments. By explicitly modeling and detecting common manipulation techniques. No single algorithm can effectively address all manipulation types. Echo Halt's integration of multiple complementary algorithms from enhanced perceptual hashing to deep feature extraction and specialized manipulation detectors creates a system that is more efficient. Real-world deployment requires careful optimization to balance detection accuracy with computational efficiency. The prototype achieves this balance through selective algorithms, decision-making process and revising minimum unique score from 70% to 40% based on real time testing and ambiguity in already established algorithms. As social media continues to evolve, the race between duplicate detection systems and evasion techniques will undoubtedly continue. The flexible, extensible architecture of Echo Halt provides a foundation for ongoing adaptation to these emerging challenges, helping maintain content integrity, protect intellectual property and improve the overall user experience on social media platforms.

REFERENCES

[1] D. Nazarov and A. Nazarov, “A fuzzy model for assessing the content’s quality impact on the growth of users on the tiktok social network,” in *2021 IEEE 23rd Conference on Business Informatics (CBI)*, vol. 2. IEEE, 2021, pp. 192–196.

[2] Iqbal, M. 2024. TikTok Revenue and Usage Statistics.

[3] Mayer, O.; Stamm, M.C. Forensic Similarity for Digital Images. IEEE Trans. Inf. Forensics Secur. 2020, 15, 1331–1346. [CrossRef]

[4] Cozzolino and Verdoliva (Cozzolino, D.; Verdoliva, L. Noiseprint: A CNN-Based Camera Model Fingerprint. IEEE Trans. Inf. Forensics Secur. 2020, 15, 144–159. [CrossRef]

[5] https: //www.facebook.com/apps/site_scraping_tos_terms.php, https://twitter.com/en/tos—accessed on 4 August 2021

[6] C. Chen, S. Ingle, and X. Lu, "Efficient multi-stage video duplicate detection based on internal video relationships," *IEEE Transactions on Multimedia*, vol. 23, pp. 2231-2244, 2020.

[7] D. Nazarov and A. Nazarov, "A fuzzy model for assessing the content's quality impact on the growth of users on the tiktok social network," in *2021 IEEE 23rd Conference on Business Informatics (CBI)*, vol. 2. IEEE, 2021, pp. 192–196.

[8] Xiao Wu, Alexander G. Hauptmann, and Chong-Wah Ngo. Practical elimination of near-duplicates from web video search. In MM, pages 218–227, 2007

[9] Y. Li, T. Zhang, and D. Tretter. 2015. An overview of video abstraction techniques. Tech. Rep. HP-2001-191, HP Laboratory, 2001.

[10] Chinmay Hedge and Venkataramana Raju. 2021. Near-Duplicate Video Detection with Deep Convolutional Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2578–2585.

[11] David G. Lowe. 2004. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60, 2 (2004), 91–110.

[12] J. Haitsma and T. Kalker. 2002. A Highly Robust Audio Fingerprinting System. In Proceedings of the International Society for Music Information Retrieval Conference (ISMIR). 107–115.

[13] T. Hoad and J. Zobel. 2003. Video similarity detection for digital rights management. In Proceedings of the 26th Australasian Computer Science Conference. 237–245.

[14] H. Jégou, M. Douze, and C. Schmid. 2011. Product Quantization for Nearest Neighbor Search. IEEE Transactions on Pattern Analysis and Machine Intelligence 33, 1 (2011), 117–128.

[15] C. Ouali, P. Dumouchel, and V. Gupta. 2015. A Spectrogram-Based Audio Fingerprinting System for Content-Based Copy Detection. Multimedia Tools and Applications 74, 2 (2015), 585–604.

[16] P. Over, G. Awad, and W. Kraaij. 2008. Content-based video retrieval evaluation: TRECVid and beyond. In Proceedings of the 30th European Conference on Information Retrieval (ECIR). 61–72.