DOI: 10.15514/ISPRAS-2025-37(6)-32

Shazam Algorithm for Partial Video Copy Detection

[1,2] *R.S. Uzdenov, ORCID: 0009-0006-5221-4950 <498rustam@gmail.com>*
[1] *A.I. Perminov, ORCID: 0000-0001-8047-0114 <perminov@ispras.ru>*

[1] *Ivannikov Institute for System Programming of the Russian Academy of Sciences,*
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.

[2] *Bauman Moscow State Technical University,*
2-ya Baumanskaya st., 5/1, Moscow, 105005, Russia.

Abstract. The Shazam algorithm has proven its reliability and efficiency in audio identification tasks. In this paper, we adapt the core principles of the Shazam algorithm for the problem of partial video copy detection. We propose a novel method for alignment video fingerprints in partial copy detection search of video query across video base. One of the best features of this method: fast CPU execution, simplicity and at the same time high efficiency. Experimental results on publicly available video datasets demonstrate that our approach achieves high accuracy in detecting partial and modified video copies, with competitive performance in terms of speed and scalability. Our findings suggest that Shazam-inspired fingerprinting can serve as an effective tool for large-scale video copy detection applications.

Keywords: partial video copy detection; shazam algorithm; perceptual hashing; keyframe extraction; video fingerprinting; nearest neighbor search; media forensics; copyright protection; large-scale video analysis; realtime detection; cpu-efficient algorithms; open-source video search.

For citation: Uzdenov R.S., Perminov A.I. Shazam Algorithm for Partial Video Copy Detection. Trudy ISP RAN/Proc. ISP RAS, vol. 37, issue 6, part 2, 2025, pp. 237-248. DOI: 10.15514/ISPRAS-2025-37(6)-32.

Алгоритм Shazam для обнаружения частичного видео копирования

[1,2] Р.Ш.Узденов, ORCID: 0009-0006-5221-4950 <498rustam@gmail.com
[1] А.И.Перминов, ORCID: 0000-0001-8047-0114 <perminov@ispras.ru>

[1] Институт системного программирования им. В.П. Иванникова РАН,
Россия, 109004, г. Москва, ул. А. Солженицына, д. 25.

[2] Московский государственный технический университет им. Н. Э. Баумана,
Россия, 105005, Москва, ул. 2-я Бауманская, 5.

Аннотация. Алгоритм Shazam доказал свою надежность и эффективность в задачах идентификации аудио. В данной работе мы адаптируем основные принципы алгоритма Shazam для задачи обнаружения частичных видеокопий. Мы предлагаем новый метод выравнивания видеоотпечатков при поиске частичной видеокопии запроса по базе видео. Одно из лучших качеств данного метода – его высокая скорость исполнения на CPU, простота и одновременно с этим высокая эффективность. Экспериментальные результаты на общедоступных видео наборах данных демонстрируют, что наш подход достигает высокой точности в обнаружении частичных и модифицированных видеокопий, обладая конкурентной производительностью по скорости и масштабируемости. Наши результаты показывают, что создание отпечатков по принципам Shazam может служить эффективным инструментом для крупномасштабных приложений по обнаружению видеокопий.

Ключевые слова: поиск копий видеофрагментов; алгоритм Shazam; перцептивное хеширование; извлечение ключевых кадров; видео фингерпринтинг; поиск ближайших соседей; медиакриминалистика; защита авторских прав; анализ видео на больших данных; обнаружение в реальном времени; ресурсоэффективные алгоритмы; открытые системы поиска видео.

Для цитирования: Узденов Р.Ш., Перминов А.И. Shazam алгоритм для обнаружения частичного видео копирования. Труды ИСП РАН, том 37, вып. 6, часть 2, 2025 г., стр. 237–248 (на английском языке). DOI: 10.15514/ISPRAS–2025–37(6)–32

1. Introduction

The rapid growth of online video content has made partial video copy detection essential for copyright enforcement, content moderation, and large-scale media search. Real-world copies are often transformed – cropped, re-encoded, overlaid, or altered in brightness, color, or speed – making robust detection challenging.

While many detection methods exist, most are either proprietary, lack open-source transparency, or depend on resource-intensive deep neural networks unsuitable for scalable, real-time processing on standard hardware. Few achieve an optimal balance of robustness, efficiency, and simplicity.

We address this gap by adapting the core principles of the Shazam algorithm [1] from audio to video. Our method extracts keyframes at regular intervals, hashes them using perceptual image hashing, and matches them via efficient approximate nearest neighbor search. Post-processing aligns candidate fragments by analyzing time differences between matches. The approach is fully parallelizable for modern CPUs (Central Processing Unit).

Evaluation on the VCDB (Video Copy Detection Benchmark) benchmark [2] demonstrates that our solution combines accuracy, efficiency, and practical scalability under diverse video transformations

The main contributions of this paper are as follows:

- We present an efficient, open-source implementation of a Shazam-inspired algorithm for partial video copy detection.
- Our pipeline achieves robust detection of copied fragments under a wide range of real-world video modifications, combining simplicity, high speed, and scalability.
- We provide comprehensive experimental results on the VCDB dataset [2], using standard metrics such as Mean Reciprocal Rank (MRR), Mean Average Precision (MAP), and recall.

2. Related Work

Video copy detection and partial video copy detection have been active research areas for more than a decade. The main approaches can be grouped into three broad categories: perceptual hashing, block-based and spatio-temporal signatures, and neural network-based methods. In addition, there are several industrial and open-source systems with varying levels of accessibility and transparency.

2.1 Surveys and Overviews

A number of survey papers have systematically reviewed the landscape of video fingerprinting and copy detection methods. Recent works, such as the 2024 survey "Digital Fingerprinting on Multimedia" (200+ references) [3], provide a taxonomy of fingerprinting techniques, covering classical block-based methods, perceptual hashes, and learning-based approaches. They highlight key requirements for real-world systems: robustness to transformations, computational efficiency, and scalability. Other recent reviews, e.g., "The 2023 Video Similarity Dataset and Challenge," [4] focus on benchmarking and the increasing role of deep learning, including transformer architectures [5], for video retrieval and similarity matching

2.1.1 Block-Based and Spatio-Temporal Signatures

Classical video fingerprinting techniques include methods based on spatial and spatio-temporal signatures. The 2009 Vobile paper [6], for example, uses spatial signatures derived from luminance (Y channel in YUV (Y component (luma) and two chroma components U and V)) and organizes video into blocks, enabling geometric transformation robustness and efficient matching. Robust ordinal measure methods [7] (2004) and TIRI (Temporally Informative Representative Images, 2009) [8] extract features from carefully sampled frames or generate 3D hashes to capture temporal context. However, such methods are often designed for full-video or coarse fragment detection, may not be robust to all types of edits, and sometimes lack public, well-maintained code.

2.1.2 Perceptual Hashing and Open-Source Libraries

Perceptual image hashing methods are popular for their simplicity, speed, and small storage footprint. Tools like OpenCV [9] and libraries such as ImageHash [10] offer a range of hash algorithms (pHash, dHash, whash, etc.). These are widely used for frame-level fingerprinting in video search pipelines. Benchmarks indicate that while CNN-based hashes [11] can provide higher accuracy, classic hashes remain competitive for many tasks. However, none of the basic image hash functions is fully robust to geometric transformations (e.g., flipping, severe cropping).

Recent open-source projects, such as VideoHash [12] and ViDeDup [13], provide practical implementations for near-duplicate detection, mainly focusing on the whole video or image collections, but often lack support for fine-grained fragment matching and may not be robust against severe modifications.

2.1.3 Neural and Learning-Based Methods

The latest advances in partial copy detection use neural networks to learn robust video representations. Winners of the DVSC23 (Dataset of Video Similarity Challenge 2023) [4] challenges have used large transformer models and deep similarity networks, achieving top performance in complex, large-scale retrieval settings. While these approaches achieve strong robustness, they require significant computational resources and are not always feasible for lightweight, real-time scenarios.

2.2 Proprietary and Industrial Systems

Many effective commercial systems exist, including those by Vobile [6] and Microsoft's PhotoDNA [14]. These are typically proprietary and designed for industrial applications (copyright, anti-piracy, CSAM (Child Sexual Abuse Material) (detection). While they set a high standard in robustness and deployment scale, the lack of transparency and public implementation limits their use in academic or open-source projects.

2.2.1 Shazam Algorithm and Audio Fragment Alignment

The Shazam algorithm [1], though originally designed for audio fingerprinting, is notable for its speed, fragment-level matching, and alignment capabilities. Its open principles have inspired similar approaches in video, including the method presented in this paper. Audio fingerprint alignment is a key idea, enabling not just duplicate detection but robust localization of copied fragments.

2.3 Benchmarks and Datasets

Several public datasets and benchmarks support research in this area. The VCDB dataset [2] is specifically designed for partial copy detection and is widely used for evaluation, have 27 hours of video content from YouTube and MetaCafe with 9k+ pairs of similar segments. DVSC23 [4] is a newer, more complex benchmark but may exceed the scale needed for lightweight systems. Proper evaluation relies on standard metrics such as Mean Reciprocal Rank (MRR), Mean Average Precision (MAP), recall, and found original accuracy.

2.4 Summary of Practical Implications

Open-source solutions do not yet achieve a balance of simplicity, CPU efficiency, robustness to video modifications, and fragment-level search. Neural approaches deliver strong accuracy but require significant resources. Most practical solutions use image/frame hashing, with search and alignment inspired by Shazam-like approaches (see Table 1).

Table 1. Summary table of different methods/tools.

Method/Tool	Approach	Key Features / Limitations	Open Source
VideoHash [12]	64-bit video hash	Fast, low memory, no fragment support, not robust to flips	Yes
VideoDeduplication [13]	DCT + clustering	Old code, not maintained, focus on full video	Yes
Wechat CV VSC2022 [15]	Deep learning (transformer)	SOTA accuracy, high complexity, high resource requirements	Yes
pyPhotoDNA [14]	Microsoft algorithm	Proprietary, large vector, slow, robust to attacks	Partially
Vobile (industry) [6]	Block-based signature	Proprietary, robust, industrial standard	No

3. Problem Statement

The proliferation of digital video content has made the ability to automatically identify copied or reused video fragments across vast multimedia collections a problem of global importance. Partial video copy detection is a foundational technology for protecting intellectual property, supporting copyright enforcement, enabling digital rights management, and combating misinformation and illicit content distribution. As video sharing and remix culture become integral to communication, a robust and efficient solution is essential for ensuring fair use and trust in digital media.

3.1 Task Definition

Given a query video fragment Q and a large database of reference videos $\mathcal{D} = \{V_1, V_2, \ldots, V_N\}$, the goal is to:

- **Detect** all segments in $\mathcal{D}$ that contain a fragment visually similar to Q, even if the fragment has been altered (cropped, re-encoded, overlaid, color modified, etc.).
- **Localize** the temporal boundaries of each detected copy, i.e., determine the start and end times within the reference video where the copy occurs.

3.2 Formal Statement

Let Q be a query video fragment of arbitrary length, and V_i a video from the reference database. The system must identify all tuples $(V_i,\ t_{start},\ t_{end})$such that the segment $V_i[t_{start}: t_{end}]$ is a copy of Q (or a substantially similar transformation), subject to some robustness threshold.

3.3 Challenges

- **Transformations:** Real-world copies may undergo various visual and temporal modifications (cropping, color jitter, overlays, speed changes, compression, geometric transforms, etc.).
- **Scalability:** The reference database may contain millions of videos, requiring solutions that are both memory- and CPU-efficient.
- **Fragment Alignment:** It is insufficient to detect only the presence of a copy; precise temporal alignment is needed to localize the copied fragment.

3.4 Evaluation Metrics

Performance is typically measured using:

- **Mean Reciprocal Rank (MRR):** Measures how quickly a relevant video is retrieved.
- **Mean Average Precision (mAP):** Captures overall ranking quality.
- **Recall:** Fraction of true copies correctly detected.
- **Found Original:** Whether the exact original is retrieved and localized.

A practical solution must achieve high accuracy across these metrics under diverse transformation conditions, while maintaining computational efficiency suitable for real-time or large-scale applications.

4. Proposed Method

This section details the proposed approach for partial video copy detection, inspired by the alignment and fingerprinting techniques of the Shazam audio algorithm [1]. The method is designed to be computationally efficient, scalable to large video collections, and robust to typical video transformations. The pipeline consists of four key stages:

1. **Keyframe Extraction**: Systematic sampling of representative frames from video sequences.
2. **Fingerprinting and Hashing**: Generation of compact, perceptual descriptors for each keyframe.
3. **Search and Alignment**: Fast identification and temporal alignment of matching fragments within a reference database.

4.1 Keyframe Extraction

A fundamental component of the proposed method is the systematic extraction of representative keyframes from each video. Keyframe extraction serves two principal goals: reducing data redundancy and enabling robust comparison between videos based on compact, meaningful content descriptors.

For both reference and query videos, frames are sampled at regular temporal intervals, typically every 0.5 seconds. This uniform sampling strategy provides a balance between computational efficiency and the ability to capture temporal variations, even in videos with scene changes or edits. The use of regular intervals, rather than scene-change detection, ensures that the approach is simple, reproducible, and does not depend on the availability or accuracy of more complex scene detection algorithms. This procedure is illustrated in Fig. 1 (first cycle), which presents the flowchart for keyframe extraction and preprocessing. Keyframes extracted at this stage are used for both constructing the reference fingerprint database and for processing query video fragments.

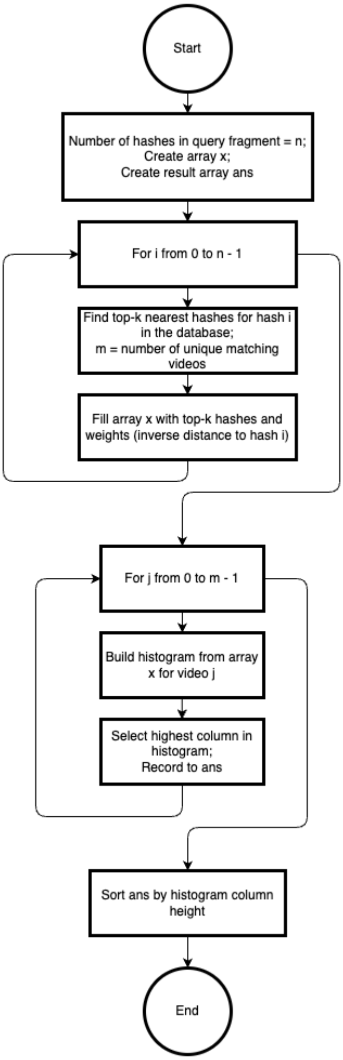
Start
Number of hashes in query fragment = n;
Create array x;
Create result array ans
For i from 0 to n - 1
Find top-k nearest hashes for hash i in the database;
m = number of unique matching videos
Fill array x with top-k hashes and weights (inverse distance to hash i)
For j from 0 to m - 1
Build histogram from array x for video j
Select highest column in histogram;
Record to ans
Sort ans by histogram column height
End

Fig. 1. Pipeline of indexing and search video.

4.2 Fingerprinting and Hashing

Following keyframe extraction, each keyframe is transformed into a compact and robust fingerprint using perceptual image hashing algorithms. The goal of this stage is to generate a representation that is both discriminative-enabling distinction between different content– and robust to common video transformations such as compression, resizing, or moderate changes in color and brightness.

For every extracted keyframe, a hash function (e.g., pHash, dHash, or other perceptual hash) is applied to produce a fixed-length descriptor. The choice of hash function is guided by the need for:

- **Robustness** to minor and color image modifications;
- **Compactness** for efficient storage and fast search;
- **Simplicity** for reproducible implementation on standard hardware.

Each hash value is associated with the corresponding video identifier and the timestamp of the keyframe. In the context of reference videos, these triples (video ID, timestamp, hash) are stored in the fingerprint database. For a query video, the resulting sequence of hashes is used as the search template.

This hashing procedure enables scalable similarity search by mapping keyframes into a common feature space where visually similar frames are close under a chosen distance metric (e.g., Hamming or Euclidean distance). The use of perceptual hashing allows the system to tolerate small transformations and degradations, which are frequently encountered in user-generated or re-encoded video content.

This stage is illustrated in the middle part of Fig. 1, where each keyframe is processed by the hash function, and the resulting fingerprint is recorded for subsequent matching and alignment.

4.3 Search and Alignment

The search and alignment stage is responsible for identifying and temporally localizing potential video copies within a large reference database. This is achieved through efficient similarity search and a robust alignment procedure inspired by the Shazam algorithm for audio [1].

4.3.1 Similarity Search

For each hash in the query sequence, the system performs a nearest neighbor search in the fingerprint database, typically retrieving the top-k closest matches based on a chosen distance metric (e.g., Hamming or Euclidean distance). Efficient indexing structures, such as Annoy, are used to enable sublinear search time even for millions of fingerprints. Each database match provides not only a candidate video ID but also the timestamp of the matched keyframe.

4.3.2 Temporal Alignment

To identify true fragment-level copies and distinguish them from incidental matches, the method aggregates all retrieved matches by calculating the time offset between the position of the query frame and the corresponding database frame. For each candidate reference video, a histogram of these offsets is constructed. Peaks in the histogram represent time shifts where multiple query frames align consistently with the same region of a reference video– strong evidence of a copied segment.

4.3.3 Fragment Localization and Ranking

The location of the highest peak in the offset histogram determines the estimated start time of the copied segment in the reference video. The height of the peak (the weighted sum of matching frames at the same offset) is used as a confidence score. Candidate videos are ranked according to this score, and the corresponding time intervals are reported as the detected copy locations.

This approach is robust to missing or noisy matches, as the alignment mechanism accumulates evidence over multiple frames. It further enables partial and modified copies to be detected and localized with high precision, even in the presence of typical video transformations.

The overall search and alignment process is visualized in Fig. 1, illustrating the construction of the offset histogram and the identification of the optimal alignment between the query and reference video sequences.

5. Experimental Setup

Given the computational intensity of keyframe extraction, hashing, and nearest neighbor search, we employ aggressive parallelization to maximize throughput and minimize wall-clock time.

5.1 Dataset

We evaluated our method on the VCDB dataset [2], which contains annotated pairs of partially overlapping real, not simulated video fragments subjected to various transformations such as cropping, overlays, changes in brightness, color jitter, rescaling, and re-encoding.

5.2 Frame Hash Functions and Search Algorithm

The search algorithm follows a “Shazam-style” pipeline:

1. Extract and hash keyframes for all database and query videos
2. Index all database hashes using Annoy for fast ANN search.
3. For each query frame, retrieve the k=10 nearest database frames.
4. For each candidate video, build a histogram of temporal offsets between query and database frames.
5. Select the offset with the highest count as the best alignment.
6. Rank candidate videos by their peak histogram value and return the top-10.

5.3 Evaluation Protocol and Metrics

We evaluate all methods using the following metrics, computed over the full set of queries:

- Mean Reciprocal Rank (MRR): Measures the average inverse rank of the first correct result.
- Mean Average Precision at 10 (mAP@10): Average precision over the top-10 retrieved results.
- Recall@10: Fraction of relevant items retrieved in the top-10.
- Found-Original: Fraction of queries where the original (ground-truth) video is present in the top-10.

All experiments are run with fixed parameters (k=10 nearest neighbors, top-10 returned).

Unfortunately, other solutions for Partial Video Copy Detection haven't measured these metrics and execution speed. Additionally, all these methods either don't have open-sourced code or are closed solutions, which makes it difficult to test them.

6. Results

6.1 Accuracy and Robustness

Fig. 2 and Table 2 summarize the global accuracy metrics for different perceptual hashing algorithms in the Shazam-based video copy detection pipeline. The results include Mean Reciprocal Rank (MRR), mean Average Precision (mAP), recall, and the fraction of queries where the original was found. All tested hash functions demonstrate competitive performance, with most achieving recall and “found original” rates above 0.9. The differences among methods are minor, indicating that even computationally efficient hashes can provide strong detection accuracy. This supports the method’s suitability for large-scale deployments where both speed and accuracy are required.

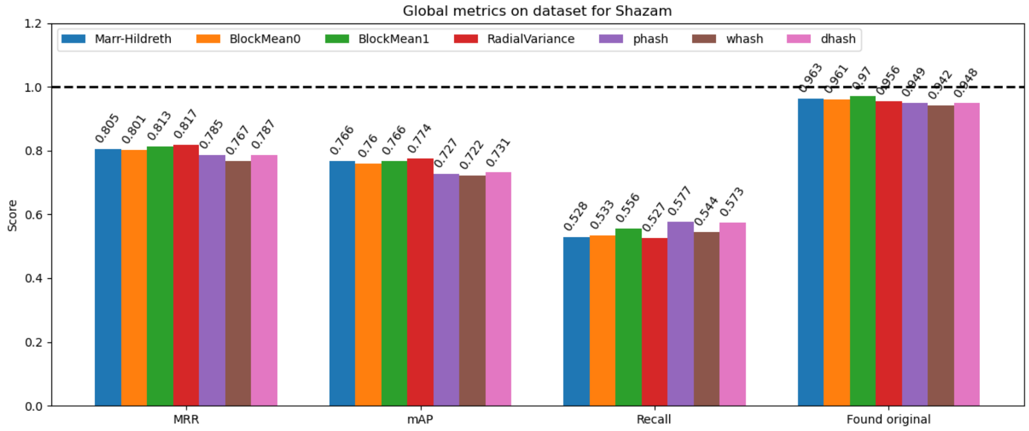

Fig.2. Main metrics across different hash algorithms.

Table 2. Results of accuracy of different hash functions. The best result is presented in bold text.

Hash Name	MRR	mAP	Recall	Found original
Marr-Hildreth	0.805	0.766	0.528	0.963
BlockMean0	0.801	0.760	0.533	0.961
BlockMean1	0.813	0.766	0.556	**0.970**
RadialVariance	**0.817**	**0.774**	0.527	0.956
phash	0.785	0.727	**0.577**	0.949
whash	0.767	0.722	0.544	0.942
dhash	0.787	0.731	0.573	0.948

6.2 Execution Speed and Scalability

The pipeline is optimized for parallel execution. With 14 parallel processes, database construction and query search achieved a 2.9–3.0× speedup compared to single-process operation. Fig. 3 and Table 3 present the average search and hash computation times for different perceptual hashing algorithms in seconds per one-hour video. The orange bars represent the mean hash computation time per algorithm, while the blue bars show the mean search time required to match video fragments using the Shazam-inspired method. Notably, algorithms such as Marr-Hildreth and whash exhibit the highest hash computation times (over 70 and 80 seconds, respectively), whereas simpler hash functions like BlockMean0 and BlockMean1 achieve significantly faster hashing performance. Across all algorithms, mean search time remains low (generally below 7 seconds), confirming the scalability and efficiency of the search stage regardless of hash type. Also, Key Frame Extractor (KFE) executed in around 450 seconds. This demonstrates that the proposed system can efficiently process long videos and large datasets, with the primary computational bottleneck attributable to the choice of hash function rather than the search phase.

7. Future Work

To address the above limitations and further enhance the system's capabilities, future work will focus on:

- **Integration of Learning-Based Features.** Augmenting the pipeline with deep neural descriptors or transformers trained for video similarity may improve robustness to challenging transformations and nontrivial copies.
- **Multimodal Fusion.** Combining visual fingerprints with audio, text, or metadata-based signatures to reduce ambiguity and improve detection in noisy or complex scenarios.

- **Adaptive Parameter Selection.** Developing automatic tuning or meta-learning strategies to select optimal parameters based on data characteristics or operational constraints.
- **Advanced Indexing and Distributed Search.** Exploring scalable indexing techniques, such as hierarchical vector search or distributed hash tables, to support petabyte-scale video archives.
- **Real-World Deployment and User Studies.** Collaborating with industry and the research community to deploy the system in practical applications, gather user feedback, and refine the approach for diverse real-world environments.

By addressing these directions, the method can become an even more powerful tool for the global community, facilitating responsible media management, copyright protection, and digital trust.

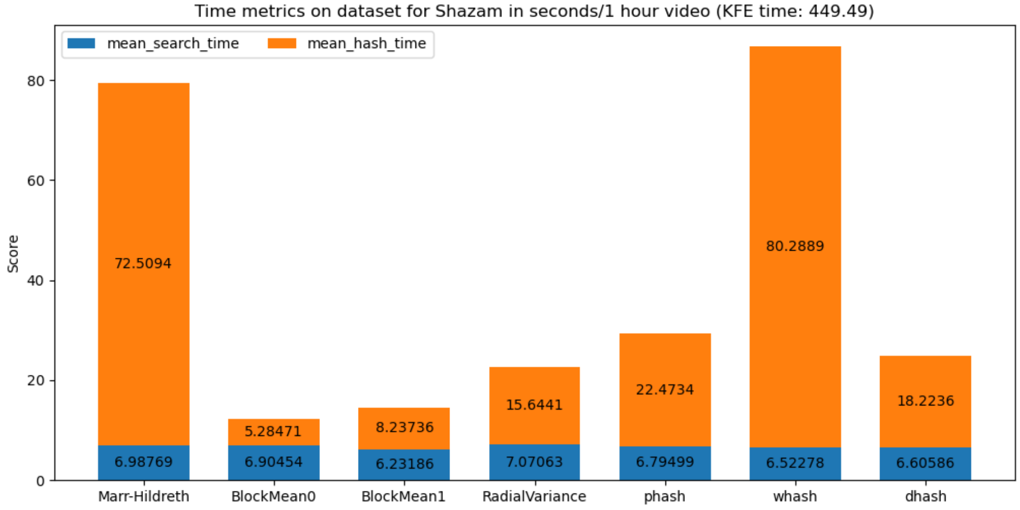

Fig.3. Speed execution of different hash algorithms in seconds per 1 hour video (on 28-core CPU).

Table 3. Results of speed comparison of different hash algorithms. The best result is presented in bold text.

Hash Name	Mean Search Time (s)	Mean Hash Time (s)
Marr-Hildreth	6.98769	72.5094
BlockMean0	**5.28471**	**6.09451**
BlockMean1	8.23796	6.23136
RadialVariance	7.07063	15.6441
phash	6.79499	22.4734
whash	6.52278	80.2889
dhash	6.05098	18.2236

8. Conclusion

In this work, we have introduced a novel method for partial video copy detection, inspired by the proven principles of the Shazam algorithm [1] in audio identification. Our approach combines systematic keyframe extraction, robust perceptual hashing, and efficient search and temporal alignment to address the challenges of detecting copied video fragments within large-scale collections. Code is available as open-source solution [16].

Extensive evaluation on the VCDB benchmark [2] confirms that the proposed method achieves high accuracy, robustness to common video transformations, and real-world scalability through CPU-parallel processing. The pipeline remains accessible and reproducible, relying solely on open-source tools and standard hardware, making it suitable for broad adoption in scientific, industrial, and societal contexts.

By enabling reliable detection and localization of video copies, this work contributes a transparent and effective solution for content protection, digital rights management, and responsible media use–serving the needs of all humanity in the evolving digital landscape. Future research will explore further improvements in robustness, scalability, and multimodal integration to extend the capabilities of this framework.

References

[1]. Wang A. et al. An industrial strength audio search algorithm // Ismir. 2003. Vol. 2003, pp. 7-13.

[2]. Jiang Y. G., Jiang Y., Wang J. VCDB: a large-scale database for partial copy detection in videos // Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13. Springer International Publishing, 2014, pp. 357-371.

[3]. Chen W., Gan W., Yu P. S. Digital Fingerprinting on Multimedia: A Survey // arXiv preprint arXiv:2408.14155. 2024.

[4]. Pizzi E. et al. The 2023 video similarity dataset and challenge // Computer Vision and Image Understanding. – 2024. Vol. 243, pp. 103997.

[5]. Vaswani A. et al. Attention is all you need // Advances in neural information processing systems. – 2017. Vol. 30.

[6]. Lu J. Video fingerprinting for copy identification: from research to industry applications // Media Forensics and Security. 2009. Vol. 7254, pp. 725402.

[7]. Hua X. S., Chen X., Zhang H. J. Robust video signature based on ordinal measure // 2004 International Conference on Image Processing, 2004. ICIP'04. IEEE, 2004. Vol. 1, pp. 685-688.

[8]. Malekesmaeili M., Fatourechi M., Ward R. K. Video copy detection using temporally informative representative images // 2009 International Conference on Machine Learning and Applications. IEEE, 2009, pp. 69-74.

[9]. Bradski G. The opencv library // Dr. Dobb's Journal: Software Tools for the Professional Programmer. 2000. Vol. 25, issue 11, pp. 120-123.

[10]. Buchner J. Image Hash library [Online] https://github.com/jgraving/imagehash (accessed 20.04.2025).

[11]. Jain T. et al. Imagededup [Online] https://github.com/idealo/imagededup (accessed 20.04.2025).

[12]. Mahanty A. Videohash [Online] https://github.com/akamhy/videohash (accessed 20.04.2025).

[13]. Katiyar A., Weissman J. {ViDeDup}: An {Application-Aware} Framework for Video De-duplication // 3rd Workshop on Hot Topics in Storage and File Systems (HotStorage 11). 2011.

[14]. Steinebach M. An analysis of photodna // Proceedings of the 18th International Conference on Availability, Reliability and Security. 2023, pp. 1-8.

[15]. Liu Z. et al. A similarity alignment model for video copy segment matching // arXiv preprint arXiv:2305.15679. 2023.

[16]. https://github.com/SUPERustam/frame_video_search (accessed 25.08.2025).

Информация об авторах / Information about authors

Рустам Шамилевич УЗДЕНОВ является студентом факультета Фундаментальных Наук МГТУ им. Баумана, лаборант института системного программирования РАН. Научные интересы: нейросетевая обработка данных, цифровая обработка изображений, методы доверенного искусственного интеллекта.

Rustam Shamilevich UZDENOV – a undergraduate student of the Faculty of Fundamental Sciences at Bauman Moscow State Technical University, and a laboratory assistant at the Institute of System Programming of the RAS. Research interests: neural network data processing, digital image processing, trust artificial intelligence.

Андрей Игоревич ПЕРМИНОВ – аспирант института системного программирования РАН. Научные интересы: нейросетевая обработка данных, цифровая обработка изображений, методы доверенного искусственного интеллекта.

Andrey Igorevich PERMINOV – a postgraduate student at the Institute of System Programming of the RAS. Research interests: neural network data processing, digital image processing, trust artificial intelligence.