A Web-based Identification Method for Illegal Streaming Videos Using Low-frequency Components of the Fast Fourier Transform

Injae Yoo, Byeongchan Park, Seok-Yoon Kim
and Youngmo Kim*

Department of Computer Science & Engineering, Soongsil University, Korea
E-mail: halo8024@outlook.com; pbc866@ssu.ac.kr; ksy@ssu.ac.kr;
ymkim828@ssu.ac.kr
**Corresponding Author*

Received 05 May 2025; Accepted 29 June 2025

Abstract

With the proliferation of web-based content platforms, the distribution of illegally streamed videos poses a serious threat to the reliability of web applications and the integrity of content copyright protection systems. Traditional video identification methods typically require the processing of large-scale feature data, which hinders the real-time performance, lightweight nature, and scalability demanded by web environments. In this paper, we propose a method for identifying illegally streamed videos that is optimized for efficient operation within web systems. The proposed approach utilizes only the low-frequency components of the fast Fourier transform (FFT). By transforming video frames into the frequency domain and extracting the structurally significant low-frequency components, the method replaces high-dimensional feature data with more compact representations. This allows the system to maintain low computational complexity and fast response times, even in web application environments. Experimental results demonstrate that, compared

Journal of Web Engineering, Vol. 24_6, 851–870.
doi: 10.13052/jwe1540-9589.2461

to existing methods, the proposed technique achieves up to 93 times reduction in feature data size, a recognition rate of 98%, and an average response time of 1745 ms. From the perspective of web engineering, the proposed method holds strong potential as a real-time identification module in web-based copyright protection systems. It offers a balanced approach that satisfies both lightweight processing requirements and high accuracy.

Keywords: Illegal streaming video, low-frequency component, fast Fourier transform, real-time video identification, copyright protection.

1 Introduction

With the recent advancement of web technologies, expansion of cloud infrastructure, and the growing popularity of various streaming-based content services, the global consumption patterns of digital content are undergoing rapid transformation. This shift is largely driven by over-the-top (OTT) and live broadcasting platforms such as Netflix, YouTube, and Twitch, which primarily rely on real-time delivery via web-based streaming. However, behind these technological developments lies a serious side effect: the surge in illegal streaming content. These unauthorized services distribute copyrighted videos without permission, causing significant financial losses to content creators and platform providers, and posing threats to the reliability and security of web-based application environments. As of 2024, reports indicate that over 20% of global internet traffic is used for the transmission of illegal content, with particularly high rates observed in the Asia-Pacific and North American regions. Most of these illegal streaming activities are conducted through the web, using highly organized systems that include user interfaces (UIs), streaming transmission mechanisms, and traffic obfuscation techniques. Consequently, technologies designed to detect and block such content must also be restructured to align with the unique demands of web environments. Conventional video identification technologies typically work by extracting keypoints from video frames, converting these into feature vectors, and comparing them with those of original content to calculate similarity. While effective at capturing local patterns, edges, and contours in images, these methods generate hundreds of high-dimensional feature descriptors per frame. This requires high-performance computing hardware and large-scale storage, making them poorly suited for real-time systems in resource-constrained web environments. This issue is particularly critical in the field of web engineering, which goes beyond simple web design to address quality,

maintainability, scalability, and performance of web-based systems through a structured approach. Therefore, illegal video identification systems must not only ensure accurate detection, but also meet demands for lightweight processing, low latency, cross-device and cross-browser compatibility, and real-time responsiveness. These requirements conflict with the computational complexity, server load, and bandwidth inefficiencies found in traditional video recognition algorithms. To address these challenges, this paper proposes a web-optimized video identification algorithm that selectively extracts only the low-frequency components of the fast Fourier transform (FFT) to concisely represent the overall structure of video frames. The Fourier transform converts spatial image data into the frequency domain, enabling more compact analysis of global patterns and changes. The low-frequency components, in particular, contain key information about an image's overall structure and contours, allowing for effective identification even with low-dimensional feature vectors. This approach outperforms conventional keypoint-based methods in terms of processing speed and storage efficiency, making it highly applicable for integration into actual web services.

The remainder of this paper is organized as follows. Section 2 reviews related work, analyzing existing video recognition algorithms and Fourier-based feature extraction methods, and highlights their limitations. Section 3 introduces the structure and processing flow of the proposed low-frequency-based identification algorithm. Section 4 presents the experimental setup and performance comparison results. Finally, Section 5 concludes the paper.

2 Related Work

2.1 Traditional Feature Extraction Methods

Traditional feature extraction techniques identify keypoints from videos or images and utilize them to compare similar images.

AKAZE extracts features using nonlinear diffusion, as illustrated in Figure 1.

The AKAZE feature extraction process employs the second-order derivatives of the image to capture overall structure and patterns. The amount of feature data generated in AKAZE depends on the number of features extracted from the scale space and increases for high-resolution images. To efficiently handle such data, a nonlinear diffusion technique is applied.

SIFT extracts scale-invariant features from images. It uses the difference of Gaussians (DoG) method to detect keypoints across multiple scales, as shown in Figure 2.

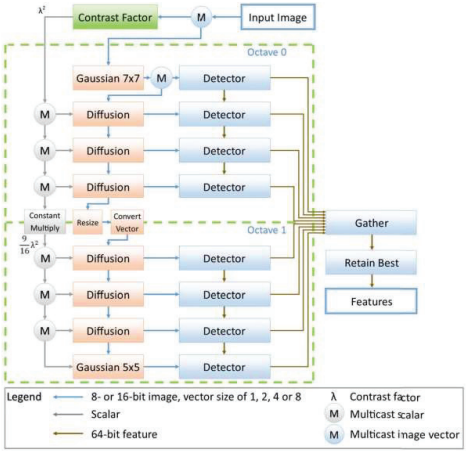

Figure 1 AKAZE (accelerated KAZE).

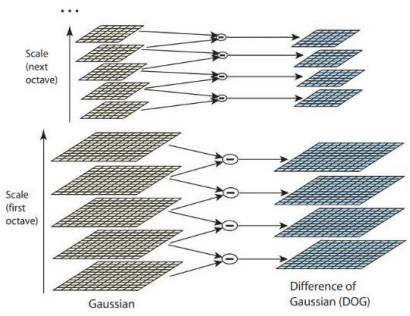

Figure 2 SIFT (scale-invariant feature transform).

Each keypoint is characterized by its position, scale, and orientation, from which a unique feature vector is generated. SIFT combines Gaussian functions and their differences to detect features at multiple resolutions. While SIFT provides high accuracy, it generates a large amount of data, leading to slower performance on large-scale datasets.

SURF, shown in Figure 3, improves upon SIFT by enhancing computational efficiency. It extracts features using the Hessian matrix and maintains rotation and scale invariance while offering faster performance than SIFT. SURF uses lower-dimensional vectors compared to SIFT, thereby increasing computational efficiency.

ORB is a feature extraction algorithm that combines the FAST keypoint detector with BRIEF binary descriptors, emphasizing speed and computational efficiency. ORB maintains rotation invariance and extracts

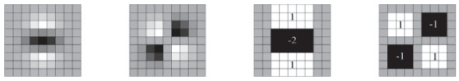

Figure 3 SURF (speeded-up robust features).

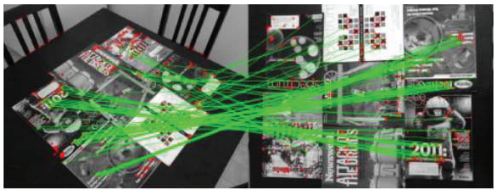

Figure 4 ORB (oriented FAST and rotated BRIEF).

features in the form of 32-dimensional binary vectors, resulting in significantly reduced data compared to other algorithms. Figure 4 illustrates the process of feature extraction and matching using ORB.

SIFT and SURF generate 128- or 64-dimensional vectors per keypoint, which require substantial storage and computation when processing many features. ORB, on the other hand, uses compact 32-dimensional binary vectors, allowing for faster processing. AKAZE uses nonlinear diffusion to extract features across multiple resolutions, which can lead to a large volume of data.

2.2 Fast Fourier Transform (FFT)

The fast Fourier transform (FFT) can be utilized to analyze image content in the frequency domain and improve image restoration performance. Both high- and low-frequency components are extracted using FFT and combined with a transformer-based architecture to enhance restoration accuracy.

Given an image $f(x, y)$, the FFT is applied to transform it into the frequency domain $F(u,\ v)$, as defined by Equation (1)

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cdot e^{-j2(\frac{ux}{M} + \frac{vy}{N})} \tag{1}$$

where M and N represent the dimensions of the image, $f(x, y)$ is the original spatial domain image, and $F(u, v)$, is its frequency domain representation, u and v denote frequency components, and j is the imaginary unit.

In video recognition, low-frequency components, which reflect the global structure and large-scale patterns of an image, are particularly useful. These

are extracted using a threshold-based filter, as shown in Equation (2)

$$F_{low}(u,v) = \begin{cases} F(u,v), & if\ \sqrt{u^2+v^2} < D \\ 0, & otherwise \end{cases} \tag{2}$$

Here, D is the cutoff threshold for low-frequency extraction, and $F_{low}(u,v)$ represents the extracted low-frequency components within a specified range. FFT enables rapid extraction of global image features by analyzing frequency components, particularly low-frequency information. This makes it a highly effective technique for illegal video detection in real-time web environments.

3 Illegal Streaming Video Identification Using Low-frequency Components of the FFT

3.1 Overview of an Illegal Streaming Video Identification Method Based on Low-frequency Components

This paper proposes an effective method for identifying illegal streaming videos by utilizing the low-frequency components of the fast Fourier transform (FFT), as illustrated in Figure 5.

The proposed approach involves converting each video frame into the frequency domain and selectively extracting only the low-frequency components, which capture the overall structural characteristics of the image. These components are then flattened into a feature vector, which serves as the basis for determining whether the content is illegally streamed.

This method significantly reduces the amount of feature data required compared to conventional keypoint-based algorithms, enabling fast identification with minimal processing delay, even in real-time streaming environments. Furthermore, the compact nature of the feature vectors makes the system highly suitable for deployment in resource-constrained environments such as web-based platforms and mobile devices.

The proposed method consists of two main stages:

(a) Original feature vector database construction: Frames are extracted at fixed intervals from the original video. Each frame is converted to grayscale and transformed using FFT. From the resulting frequency domain, a central 30×30 region of low-frequency components is extracted and reshaped into a one-dimensional vector. These vectors are then stored in a reference database.

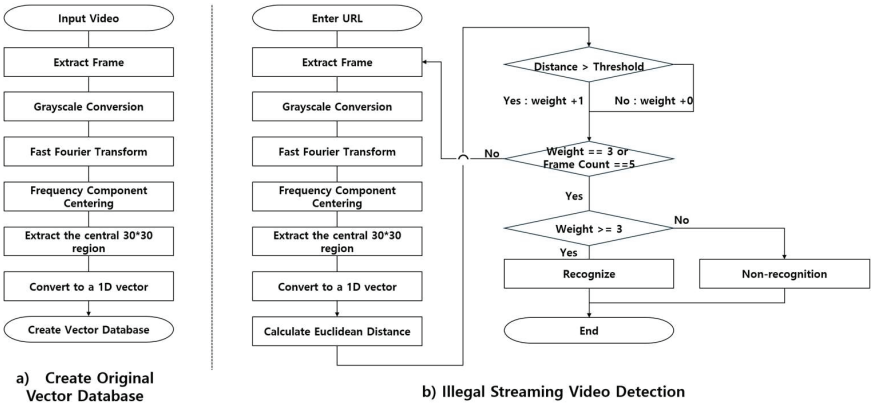

Figure 5 Illustration of the overall process flow of the proposed identification method.

(b) Illegal streaming video identification: Several frames are captured from a suspicious streaming URL. The same preprocessing and FFT-based feature extraction steps are applied. The generated vectors are then compared against those in the reference database using Euclidean distance. The number of frames whose similarity falls below a predefined threshold is used as a criterion for determining whether the video stream is illegal.

3.2 Construction of the Original Feature Vector Database

The process for constructing the original feature vector database is illustrated in Figure 5(a). First, video frames are uniformly sampled at 1 s intervals from the input video. This fixed temporal sampling is applied regardless of the video's frame rate (FPS), ensuring consistent comparison units suitable for real-time streaming environments.

To maintain uniformity in processing, each sampled frame is resized to a resolution of 256 $\times$ 256 pixels. To reduce computational complexity, the RGB channels are removed, and the image is converted into grayscale.

A 2D fast Fourier transform (FFT) is then applied to each grayscale frame, converting the image from the spatial domain to the frequency domain, as described in Equation (3). This transformation enables the extraction of frequency-based structural features from the image.

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \cdot e^{-j2(\frac{ux}{M}+\frac{vy}{N})} \tag{3}$$

Here, $f(x, y)$ denotes the pixel value in the spatial domain, while $F(u, v)$ represents the corresponding component in the frequency domain. The result of the transformation is a 256 $\times$ 256 complex-valued array, where low-frequency components are typically located in the upper-left region, and high-frequency components are located toward the bottom-right.

Since the structural characteristics of an image are primarily concentrated in the low-frequency components, the method selectively extracts only these components. This emphasizes the overall shape and contour of the frame rather than fine-grained details. The low-frequency component extraction is performed according to Equation (4).

$$F_{shifted}(u, v) = F\left(u + \frac{M}{2} mod\ M, v + \frac{N}{2} mod\ N\right) \tag{4}$$

Here, $F_{shifted}(u, v)$ refers to the frequency domain array after centering the low-frequency components, while $F(u, v)$ is the original output of the Fourier transform. The parameters M and N represent the width and height of the frequency array, respectively, and the mod operation ensures that the indices remain within valid array bounds when shifting the frequency components.

From the shifted array, a central 30 $\times$ 30 region, where the most prominent low-frequency components are concentrated, is extracted. This block is then flattened into a one-dimensional vector and stored in the original feature vector database.

Since the extracted low-frequency block contains complex values, only the magnitude is used to convert the data into a real-valued vector. The resulting 2D magnitude array is flattened into a 900-dimensional 1D vector. Finally, this vector is normalized to the range [0,1] using min-max normalization, as defined in Equation (5).

$$v' = \frac{v - \min(v)}{\max(v) - \min(v)} \tag{5}$$

3.3 Illegal Streaming Video Identification

The procedure for identifying illegal streaming videos is illustrated in Figure 5(b). This process is designed to operate efficiently without downloading the entire video, utilizing selective stream sampling. This approach addresses the limitations of network bandwidth in streaming environments and meets the requirements of real-time analysis, thereby enhancing applicability in web-based systems.

First, the system accesses the URL of the suspected illegal streaming video and downloads the playlist file (e.g., .m3u8). From the multiple media streams listed in the playlist, one is selectively downloaded, and five representative frames are extracted from the stream. The frame extraction is based on either fixed time intervals (e.g., one frame per second) or evenly spaced segments to capture the structural characteristics of the entire video.

Each extracted frame is processed using the same steps described in Section 3.2, including grayscale conversion, FFT computation, low-frequency component selection, vectorization, and normalization, resulting in five query vectors.

Each query vector is then compared with the vectors stored in the original feature vector database. The comparison uses cosine similarity, where the similarity score s between two vectors is defined by Equation (6).

$$S = \cos(\theta) = \frac{A \cdot B}{\|A\| \cdot \|B\|} \tag{6}$$

After all comparisons are completed, if three or more out of the five frames exhibit a cosine similarity of 80% or higher, the streaming video is determined to be an unauthorized replica or a highly similar illegal copy of the original video.

The 80% threshold was empirically determined as optimal in preliminary sensitivity analysis. Variations between 70–90% were tested, and 80% provided the best balance between false positives and false negatives. A more detailed sensitivity analysis is planned in future work to validate this parameter across broader datasets.

4 Experiments and Results

4.1 Experimental Environment

To evaluate the effectiveness of the proposed method for identifying illegal streaming videos using the low-frequency components of the fast Fourier transform (FFT), an experimental setup was constructed, as shown in Table 1.

4.2 Original Feature Vector Database Construction

The process for generating the original feature information vectors is illustrated in Figure 6.

Table 1 Experiment environment

		Spec
SW	OS	Windows 11 Pro 64bit
	Spec	Python 3.13
HW	spec	Intel Core i7-1360P
		32 GB
		1 TB

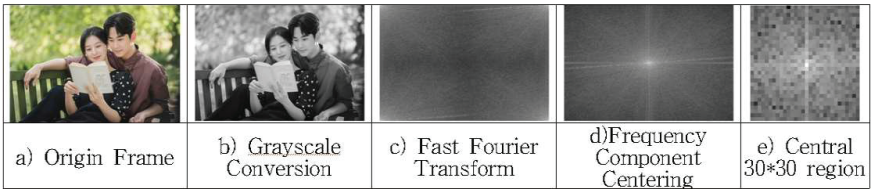

Figure 6 Method for generating feature information vector using the fast Fourier transform.

The steps are as follows:

(a) Frame extraction: Frames are sampled at fixed time intervals from the source video.

(b) Grayscale conversion: To reduce computational complexity, the extracted frames are converted from RGB to grayscale by removing color information.

(c) Fast Fourier transform (FFT): A 2D FFT is applied to each grayscale frame to transform it into the frequency domain, separating the image into frequency components. The low-frequency components are especially useful for analyzing the global structure and coarse patterns of the frame.

(d) Frequency shift: To simplify the identification of important structural information, the low-frequency components are shifted to the center of the frequency array.

(e) Low-frequency region extraction: From the shifted frequency array, the central 30×30 region – which predominantly contains low-frequency information – is extracted. This block is then used to construct the feature information vector.

4.3 Comparison of Illegal Streaming Detection Algorithms

To validate the efficiency of the proposed illegal streaming detection method based on the low-frequency components of the fast Fourier transform (FFT),

Table 2 Performance comparison results

Category	AKAZE	SURF	SIFT	ORB	FFT
Number of feature points	532	450	619	500	1
Data per feature point (bytes)	288	288	544	64	3600
Total data (bytes)	153,216	129,600	336,736	32,000	3600
Relative data amount (%)	4256	3600	9354	889	100

we conducted a comparative analysis with existing feature extraction algorithms – AKAZE, SURF, SIFT, and ORB – using a single video frame. As shown in Table 2, traditional methods require a significantly larger amount of data due to the process of detecting multiple keypoints and generating individual descriptors for each feature.

The proposed FFT-based method extracts only one feature vector per frame, consisting of 900 floating-point values, each occupying 4 bytes, resulting in a total of 3600 bytes per frame. In contrast, traditional algorithms extract hundreds of keypoints, each with its own descriptor, leading to significantly higher total data per frame – ranging from approximately 8 to 93 times more than the proposed method.

- Number of feature points refers to the total keypoints detected from a single frame.
- Data per feature point indicates the size (in bytes) of the descriptor associated with each keypoint.
- Total is the product of the number of feature points and the data size per point.
- Relative data amount expresses the total data of each method as a percentage relative to the FFT-based method (set to 100%).

This result confirms the extreme compactness and data efficiency of the FFT-based approach, making it highly suitable for real-time web-based applications.

4.4 Validation of Illegal Streaming Video Detection

To evaluate the effectiveness of the proposed illegal streaming detection method using low-frequency components of the fast Fourier transform (FFT), we conducted an experiment using 100 original videos. For each original video, a feature vector was generated using the proposed method and stored in the feature vector database. Then, query feature vectors were extracted from streaming video samples and used to test identification accuracy.

Table 3 Experimental result

Query Video	Found Video	Match Count	Recognition	Time (ms)
Test_001.mp4	Test_001.mp4	805	Recognized	1528.0195
Test_002.mp4	Test_002.mp4	895	Recognized	1795.297
Test_003.mp4	Test_003.mp4	850	Recognized	1369.389
Test_004.mp4	Test_004.mp4	856	Recognized	1720.664
Test_005.mp4	Test_005.mp4	821	Recognized	1885.338
Test_006.mp4	Test_006.mp4	770	Recognized	1965.837
Test_007.mp4	Test_007.mp4	852	Recognized	1511.089
Test_008.mp4	Test_008.mp4	871	Recognized	1491.831
Test_009.mp4	Test_009.mp4	824	Recognized	1699.84
Test_010.mp4	Test_010.mp4	837	Recognized	1455.333
Test_011.mp4	Test_011.mp4	866	Recognized	1790.147
Test_012.mp4	Test_012.mp4	849	Recognized	1710.974
Test_013.mp4	Test_013.mp4	135	Unrecognized	3157.089
Test_014.mp4	Test_014.mp4	880	Recognized	1437.473
Test_015.mp4	Test_015.mp4	899	Recognized	1510.386
Test_016.mp4	Test_016.mp4	802	Recognized	1872.183
Test_017.mp4	Test_017.mp4	751	Recognized	2006.812
Test_018.mp4	Test_018.mp4	837	Recognized	2431.168
Test_019.mp4	Test_019.mp4	787	Recognized	2231.876
Test_020.mp4	Test_020.mp4	879	Recognized	1992.12
Test_021.mp4	Test_021.mp4	770	Recognized	1832.918
Test_022.mp4	Test_022.mp4	807	Recognized	2013.989
Test_023.mp4	Test_023.mp4	771	Recognized	1561.903
Test_024.mp4	Test_024.mp4	838	Recognized	1650.15
Test_025.mp4	Test_025.mp4	798	Recognized	1300.273
Test_026.mp4	Test_026.mp4	808	Recognized	1676.346
Test_027.mp4	Test_027.mp4	764	Recognized	2033.792
Test_028.mp4	Test_028.mp4	800	Recognized	1682.092
Test_029.mp4	Test_029.mp4	857	Recognized	1512.787
Test_030.mp4	Test_030.mp4	804	Recognized	1637.067
Test_031.mp4	Test_031.mp4	813	Recognized	1962.225
Test_032.mp4	Test_032.mp4	880	Recognized	1668.271
Test_033.mp4	Test_033.mp4	800	Recognized	1999.976
Test_034.mp4	Test_034.mp4	884	Recognized	1351.603
Test_035.mp4	Test_035.mp4	770	Recognized	1483.909
Test_036.mp4	Test_036.mp4	822	Recognized	1593.007
Test_037.mp4	Test_037.mp4	767	Recognized	1352.015
Test_038.mp4	Test_038.mp4	881	Recognized	2628.099
Test_039.mp4	Test_039.mp4	838	Recognized	1416.12

(*Continued*)

Table 3 Continued

Query Video	Found Video	Match Count	Recognition	Time (ms)
Test_040.mp4	Test_040.mp4	809	Recognized	2019.465
Test_041.mp4	Test_041.mp4	763	Recognized	1545.182
Test_042.mp4	Test_042.mp4	758	Recognized	1590.864
Test_043.mp4	Test_043.mp4	243	Unrecognized	3213.557
Test_044.mp4	Test_044.mp4	802	Recognized	1307.445
Test_045.mp4	Test_045.mp4	879	Recognized	1546.179
Test_046.mp4	Test_046.mp4	833	Recognized	1701.058
Test_047.mp4	Test_047.mp4	841	Recognized	1491.195
Test_048.mp4	Test_048.mp4	860	Recognized	1498.253
Test_049.mp4	Test_049.mp4	757	Recognized	2071.018
Test_050.mp4	Test_050.mp4	784	Recognized	2046.496
Test_051.mp4	Test_051.mp4	830	Recognized	1612.47
Test_052.mp4	Test_052.mp4	799	Recognized	1719.75
Test_053.mp4	Test_053.mp4	853	Recognized	2116.105
Test_054.mp4	Test_054.mp4	881	Recognized	1317.056
Test_055.mp4	Test_055.mp4	751	Recognized	1846.407
Test_056.mp4	Test_056.mp4	883	Recognized	2337.371
Test_057.mp4	Test_057.mp4	803	Recognized	2334.51
Test_058.mp4	Test_058.mp4	855	Recognized	1163.027
Test_059.mp4	Test_059.mp4	753	Recognized	1516.212
Test_060.mp4	Test_060.mp4	803	Recognized	1795.598
Test_061.mp4	Test_061.mp4	895	Recognized	1348.437
Test_062.mp4	Test_062.mp4	793	Recognized	1544.526
Test_063.mp4	Test_063.mp4	763	Recognized	1703.813
Test_064.mp4	Test_064.mp4	844	Recognized	2165.396
Test_065.mp4	Test_065.mp4	797	Recognized	1731.379
Test_066.mp4	Test_066.mp4	764	Recognized	1810.081
Test_067.mp4	Test_067.mp4	789	Recognized	1898.727
Test_068.mp4	Test_068.mp4	831	Recognized	1908.047
Test_069.mp4	Test_069.mp4	860	Recognized	1753.43
Test_070.mp4	Test_070.mp4	802	Recognized	1385.874
Test_071.mp4	Test_071.mp4	773	Recognized	1981.902
Test_072.mp4	Test_072.mp4	873	Recognized	1874.447
Test_073.mp4	Test_073.mp4	790	Recognized	1415.974
Test_074.mp4	Test_074.mp4	764	Recognized	1540.31
Test_075.mp4	Test_075.mp4	794	Recognized	2011.253
Test_076.mp4	Test_076.mp4	814	Recognized	1926.645
Test_077.mp4	Test_077.mp4	838	Recognized	1528.911
Test_078.mp4	Test_078.mp4	820	Recognized	1664.02

(*Continued*)

Table 3 Continued

Query Video	Found Video	Match Count	Recognition	Time (ms)
Test_079.mp4	Test_079.mp4	758	Recognized	1701.961
Test_080.mp4	Test_080.mp4	837	Recognized	1930.101
Test_081.mp4	Test_081.mp4	878	Recognized	2119.682
Test_082.mp4	Test_082.mp4	885	Recognized	1764.695
Test_083.mp4	Test_083.mp4	812	Recognized	2095.762
Test_084.mp4	Test_084.mp4	888	Recognized	2006.995
Test_085.mp4	Test_085.mp4	830	Recognized	1806.695
Test_086.mp4	Test_086.mp4	885	Recognized	1311.955
Test_087.mp4	Test_087.mp4	782	Recognized	1436.687
Test_088.mp4	Test_088.mp4	872	Recognized	1967.372
Test_089.mp4	Test_089.mp4	754	Recognized	1231.841
Test_090.mp4	Test_090.mp4	790	Recognized	1780.694
Test_091.mp4	Test_091.mp4	777	Recognized	2045.146
Test_092.mp4	Test_092.mp4	884	Recognized	1790.726
Test_093.mp4	Test_093.mp4	821	Recognized	1564.581
Test_094.mp4	Test_094.mp4	761	Recognized	1491.016
Test_095.mp4	Test_095.mp4	782	Recognized	1989.846
Test_096.mp4	Test_096.mp4	797	Recognized	1886.512
Test_097.mp4	Test_097.mp4	811	Recognized	1659.353
Test_098.mp4	Test_098.mp4	786	Recognized	2064.588
Test_099.mp4	Test_099.mp4	848	Recognized	2001.199
Test_100.mp4	Test_100.mp4	270	Unrecognized	3255.238
Recognition rate	**98%**		**Recognition Time**	**1744.832 ms**

The results are summarized in Table 3. In the table:

- Query video refers to the original (authorized) content.
- Found video refers to the streaming video being queried.
- For each found video, five frames were extracted and processed using the proposed method to generate query vectors.
- A streaming video is classified as an illegal copy if three or more out of the five query frames show a cosine similarity of 80% or higher with their counterparts in the original feature vector database.

The match count indicates the number of query frames (out of 5) that matched with the original database with a similarity score exceeding the 80% threshold.

Based on the results of Table 3 and Figure 6, the proposed FFT-based identification method achieved a high recognition rate of 98% across all test videos. In addition, the average recognition time per video was less than 2 s

(1.7 s), demonstrating excellent computational efficiency suitable for real-time processing.

Notably, the method consistently produced accurate identification results through quantitative similarity-based comparison. The few failure cases (e.g., Test_100.mp4) were primarily due to extremely dissimilar structural patterns or severe frame degradation, which hindered accurate feature extraction.

Among the 100 test cases, 3 videos (Test_013, Test_043, Test_100) failed to be recognized. These cases were analyzed separately. Test_013 and Test_043 suffered from significant compression artifacts and low bitrate streaming, resulting in structural distortion. Test_100 used a different encoding scheme and non-standard resolution, altering the frequency distribution pattern. These failures highlight potential limitations in robustness under highly degraded or format-shifted conditions.

These results empirically validate the effectiveness of the proposed method for fast and accurate illegal video detection in real-world web streaming environments.

5 Conclusion

This paper proposed a lightweight and effective method for real-time identification of illegal streaming videos in web environments, utilizing the low-frequency components of the fast Fourier transform (FFT). By applying the FFT on a per-frame basis and extracting only the central low-frequency region to construct a compact one-dimensional feature vector, the method demonstrates significant advantages in processing speed and data efficiency compared to conventional high-dimensional keypoint-based approaches. Experimental results show that the proposed method accurately identified 98 out of 100 test videos, achieving a recognition rate of 98%, with an average processing time of 1744.832 ms, making it suitable for real-time web-based applications. Furthermore, the total size of the extracted feature information was reduced by up to 93 times compared to traditional methods, confirming the method's applicability across various environments such as web servers, cloud APIs, and mobile devices. From a web engineering perspective, the proposed approach satisfies key system requirements, including low latency, lightweight architecture, and scalability. Its playlist-based stream access and frame-selective processing structure are naturally compatible with modern web streaming architectures, suggesting strong potential for integration as an illegal content detection module within web platforms. However, this study has several limitations. First, recognition accuracy may decline when the

video structure is heavily distorted or if encoding formats differ significantly. Second, robustness across diverse resolutions and frame rates requires further experimentation. Lastly, extending the system to support real-time distributed processing in multi-user environments remains a topic for future work.

In future research, we plan to perform quantitative performance evaluations under various video distortion conditions, explore hybrid architectures combining deep learning with the proposed method, and investigate integration with CDN-based monitoring systems as well as browser-level feature embedding for enhanced detection capability.

Acknowledgment

This research project was supported by Ministry of Culture, Sport and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sport and Tourism in 2025 (Project Name: Development of Copyright Technology for OTT Contents Copyright Protection Technology Development and Application, Project Number: RS-1375027563, Contribution Rate: 100%).

References

[1] I. Yoo, J. Lee, S. Jang, B. Park, S. Kim, and Y. Kim, "An illegal distribution platform tracking method using IP port scanning," *Journal of Software Assessment and Valuation*, vol. 19, no. 4, pp. 115–122, 2023. DOI: 10.29056/jsav.2023.12.12.

[2] D. J. Spajic, "Piracy is back: Piracy statistics for 2024," *DataProt*, Feb. 2024. [Online]. Available: https://dataprot.net/statistics/piracy-statistics/.

[3] C. H. Kim, H. J. Yu, S. Y. Kim, and S. H. Oh, "Efficient techniques to block copyright infringement illegal streaming sites," *Journal of KIISC*, vol. 32, no. 5, pp. 837–844, 2022. DOI: 10.13089/JKIISC.2022.32.5.837.

[4] G. Yoon, Y. Lee, and S. Choi, *A Study on the Improvement of the Prevention System of Illegal Harmful Information Distribution*, Korea Communications Commission, Rep. KCC-2023-33, 2023.

[5] M. Nickel, L. Kalms, T. Haring, and D. Göhringer, "High-performance AKAZE implementation including parametrizable and generic HLS

modules," in *Proc. IEEE ASAP*, 2022, pp. 139–147. DOI: 10.1109/ASAP54787.2022.00031.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004. DOI: 10.1023/B:VISI.0000029664.99615.94.

[7] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," in *ECCV*, Lecture Notes in Computer Science, vol. 3951, Springer, Berlin, Heidelberg, 2006, pp. 404–417.

[8] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2011, pp. 2564–2571. DOI: 10.1109/ICCV.2011.6126544.

[9] S. Paul, S. Kumawat, A. Gupta, and D. Mishra, "F2former: When fractional Fourier meets deep Wiener deconvolution and selective frequency transformer for image deblurring," *arXiv preprint*, arXiv:2409.02056, 2024. DOI: 10.48550/arXiv.2409.02056.

[10] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed., Pearson, 2018.

[11] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, 1989.

[12] J. Hu, F. Jia, and W. Liu, "Application of Fast Fourier Transform," *Highlights in Science, Engineering and Technology*, vol. 38, pp. 590–595, 2023.

[13] D. Acharya, A. Billimoria, N. Srivastava, and A. Bhardwaj, "Emotion recognition using Fourier transform and genetic programming," *Applied Acoustics*, vol. 165, 107328, 2020. DOI: 10.1016/j.apacoust.2020.107328.

[14] S. Kiruthika and V. Masilamani, "Image quality assessment based fake face detection," *Multimedia Tools and Applications*, vol. 81, pp. 12345–12367, 2022. DOI: 10.1007/s11042-021-11000-z.

[15] B. Nassi, R. Ben-Netanel, A. Shamir, and Y. Elovici, "Game of Drones: Detecting streamed POI from encrypted FPV channel," in *Proc. ACM SIGSAC Conf. Computer and Communications Security*, 2018, pp. 123–137. DOI: 10.1145/3243734.3243783.

[16] Y. Zhao and P. Krähenbühl, "Real-time online video detection with temporal smoothing transformers," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 1234–1243. DOI: 10.1109/CVPR52688.2022.00131.

Biographies

Injae Yoo received his Bachelor's degree in software engineering from The Cyber University of Korea in 2017, his Master's degree in computer science and engineering from Soongsil University in 2022, and is currently pursuing a Ph.D. in computer science and engineering at Soongsil University. His research interests include lightweight video analysis, illegal streaming detection, and real-time web-based identification systems.

Byeongchan Park received his Bachelor's degree in 2015, his Master's degree in computer engineering from Soongsil University in 2018, and his doctorate in computer engineering from Soongsil University in 2023. His research interests include copyright protection and utilization activation.

Seok-Yoon Kim received his B.Sc. degree in electrical engineering from Seoul National University in 1980, his M.Sc. degree in ECE from the University of Texas at Austin in 1990, and his Ph.D. degree in ECE from the University of Texas at Austin in 1993. His research interests include system design methodology and copyright protection technology.

Youngmo Kim received his Bachelor's degree in computer engineering from Daejeon University in 2003, his Master's degree in computer engineering from Daejeon University in 2005, and his doctorate in computer engineering from Daejeon University in 2011. His research interests include copyright protection and utilization activation.