**Fractal Geometry-Based Dynamic Watermarking for Intellectual Property Protection in Multimedia Content**

**Abstract**

With rising threats of intellectual property theft, piracy, and copyright violations in multimedia content, there is an increasing need for intelligent and secure protection mechanisms. Traditional static watermarking methods such as CS-SCHT, SURF, DNN, 2D-DFT, and LWT, which embed fixed information across the entire media, are easily detectable, lack adaptability to content variations, and remain highly vulnerable to removal or tampering, thereby limiting their effectiveness. To address these challenges, a fragile and dynamic watermarking approach is proposed, which continuously adapts to video content and employs fractal geometry-based Sierpinski triangle embedding to achieve robust, imperceptible, and tamper-sensitive watermark integration. This method ensures video protection and copyright enforcement by leveraging object detection models such as YOLOv11. The Sierpinski triangle patterns, based on self-similarity and fractal complexity, ensure imperceptibility while enabling reliable modification detection, thereby supporting tracking, traceability, ownership verification, and real-time adaptability without degrading visual quality. The effectiveness of the proposed approach is validated using standard performance metrics, with results including Hamming Distance of 3.32, PSNR of 46 dB, BER of 0.01, NC of 0.99, mAP of 0.84, and ET of 2.96 confirming high imperceptibility and sensitivity to tampering across diverse video scenarios, while significantly outperforming traditional static watermarking methods.

**Keywords**

Dynamic watermarking, IPR theft, piracy, security system, multimedia protection, YOLOv11, PLSB, Sierpinski triangle patterns.

**1. Introduction**

The growth of digital video content has raised serious concerns about video authenticity and fake content detection [1]. Deepfake technology allows people to create realistic but fake videos, making it difficult to identify what is real and what is modified [2]. Current watermarking methods add fixed information to videos for copyright protection, but they cannot effectively detect if videos have been tampered with or faked [3]. Modern computer vision and artificial intelligence offer new ways to analyse video content and protect it from unauthorized changes [4]. Digital forensics needs reliable methods to embed tracking information and detect unauthorized video modifications to prove content ownership [5]. This research develops a new video authentication system that combines smart content analysis with watermarking techniques to detect fake videos and unauthorized changes through embedded information verification.

**a. Motivation**

Digital video piracy and deepfake-based falsification are escalating threats, with deepfakes increasing by over 550% from 2019 to 2023 [3]. These challenges hinder cyber forensic investigations and contribute to massive IPR theft. This calls for intelligent, adaptive watermarking to ensure authenticity, traceability, and ownership protection in digital content.

**b. Work Contributions**

The work contributions of our work are as follows:

* To detect objects in videos using YOLOv11 for content analysis.
* To embed dynamic watermarks containing IP address, timestamp, location, and video hash using PLSB technique.
* To extract watermarks and verify video authenticity by comparing hash values.
* To validate the system's ability to identify deepfake and tampered videos through watermark verification.

**c. Notations and symbols**

Notations are represented by an identifier, sign, word arrangement of characters or abbreviated expressions. A mark, or term that indicates an object is referred to as a symbol. These notations are clearly described in Table 2.

**d. Structure of the paper**

Section 2 pertains to the existing approaches toward correlated work and trend analysis. In Section 3, it gives detailed information and described the overall workflow. Section 4 is all about the results and execution analysis of the system. In section 5 provides the Performance and analysis of the system. Finally, section 6 presents summary of the work.

**2. Literature survey**

Aouthu et al. [5] proposed a blind video watermarking technique that integrates CS-SCHT and SURF for robust content protection. CS-SCHT ensures resilience against compression, noise, and histogram equalization. SURF enables correction of geometric attacks such as scaling, rotation, translation, and illumination changes using affine transformation. The watermark is embedded in low-frequency phase components to preserve visual quality. However, the method fails to recover the watermark if SURF fails to extract sufficient matching feature points under severe distortion.

Yang et al. [6] proposed a robust video watermarking algorithm based on the two-dimensional discrete Fourier transform (2D-DFT) to enhance resistance against common signal processing and geometric attacks. A limitation of the method is that it does not integrate advanced security features like blockchain or decentralized storage, which may limit its application in tamper-proof video authentication scenarios.

Singh et al. [7] proposed a lossless video watermarking technique using intelligent keyframe selection via the IGSA algorithm in the LWT domain. A scrambled watermark logo is embedded into motion frames after one-level transformation, aiming to protect content with minimal perceptual distortion. However, since the method relies heavily on motion-based keyframes, it may not perform effectively for videos with minimal or no motion content.

T. Jayamalar et al. [8]. proposed a CNN-based watermarking method for medical data protection, using a deep neural network and the Levenberg–Marquardt algorithm to balance robustness and invisibility. Though effective, the approach uses static watermarking, which lacks adaptability and is more prone to detection or tampering.

Hsu et al. [9] proposed an AI-assisted deepfake detection approach using adaptive blind image watermarking as the core technology to embed watermarks imperceptibly and verify image authenticity without needing the original image for extraction. A limitation of the method is that its robustness may decrease against highly sophisticated deepfake techniques that can adaptively distort or remove watermarks.

Table 1. Summary of the literature survey

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Author** | **Watermark Adaptivity** | **ML Model** | **Embedding Technique** | **Watermark type** | **Security Features** | **Application** |
| Aouthu et al. [5] | Static | N/A | CS-SCHT, SURF | Robust | N/A | Video integrity & piracy |
| Yang et al. [6] | Static | N/A | 2D-DFT | Robust | N/A | Copyright protection |
| Singh et al. [7] | Dynamic | GSA | LWT | Robust | Scrambling | Copyright protection |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Jayamalar et al. [8] |  |  |  |  |  | | Static | DNN, Levenberg-Marquardt optimization | DNN | Robust | Neural watermarking | IPR verification |
| L.-Y. Hsu et al. [9] | Dynamic | Mixed Modulation, GWO, DAE | Custom blind watermarking using mean-based adaptive embedding | Fragile | Blind extraction | Deepfake detection |
| S. J. Amruth et al. [10] | Static | N/A | Fourier Transform | Robust | Robustness | Ownership protection |
| Proposed work | Dynamic | YoloV8, PLSB | Dynamic | Fragile | Context-bound watermark using user/video metadata+ SHA-256 binding | Piracy & Copyright protection |

**3. Proposed Methodology**

This section outlines the proposed fragile and dynamic watermarking framework designed for video content authentication and intellectual property (IP) protection. As shown in Fig. 1, the methodology is structured into three principal modules: 1) Object Localization using YOLOv11, 2) Fractal-Based Adaptive Watermark Embedding and 3) Watermark Extraction and Authenticity Validation. Table II provides a clear description of the symbols and notations adopted in presenting the proposed methodology.

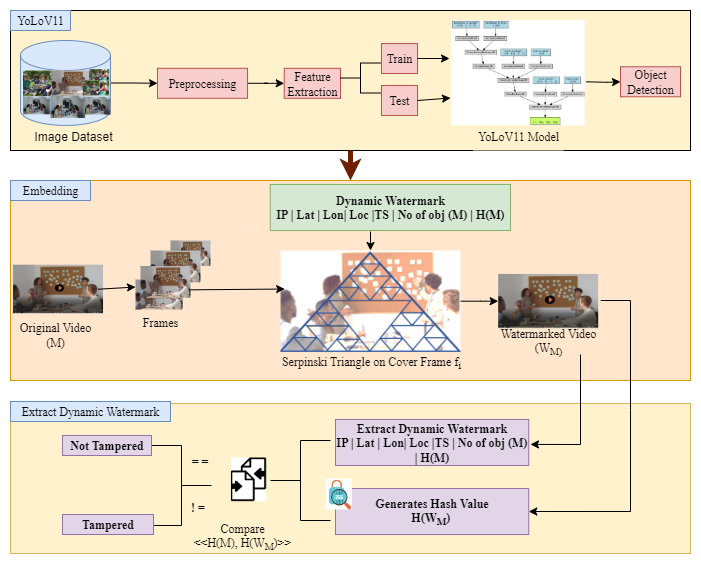
****

Fig. 1. Proposed Architecture

Table 2. Notations

|  |  |
| --- | --- |
| Symbol | Description |
| M | Input video sequence |
|  | Frame t of video |
| n | Total number of frames in M |
| H, W | Original frame height and width |
|  | Network input height and width |
|  | Resized frame t |
|  | Pre-processed video sequence |
|  | Normalized Frame t |
|  | Feature map at backbone layer l |
|  | |  | | --- | | Convolutional kernel weights | |
|  | Bias term |
| \* | Convolution Operator |
| L | Total number of convolutional layers |
|  | Extracted feature map at stride si for frame t |
|  | Stride values (8,16,32) |
|  | Linearly transformed feature map at scale si for frame t |
|  | Trainable weight of neck linear transformation for scale i |
|  | Trainable bias of neck linear transformation for scale i |
|  | Linear transformation operator |
| FFM | Feature Fusion Module |
|  | Co-ordinates of predicted box |
|  | Width and height of predicted box |
|  | Localization Loss |
|  | objectness loss |
|  | classification loss |
|  | Total training loss |
|  | Overlay operator |

*3.1 Object Localization using YOLOv11*

YOLOv11 is a one-stage object detection framework, employed in this work solely as a backbone for feature extraction. While it is typically trained and evaluated on the COCO dataset, which consists of over 330,000 images with 80 annotated object categories, The input is a video associated with is considered where each frame Ft​ ∈ RH×W×3 denotes an RGB image of Height H, Width W and n frames.

Each frame is first resized to the fixed network resolution (H′×W′) using an interpolation operator R (.), expressed as with . After resizing, normalization is applied by scaling the pixel intensities into [0,1] as shown in Eq. (1). Consequently, the complete pre-processed sequence can be represented as , which forms the standardized input to the backbone.

, Eq. (1)

Each Normalized frame is passed to the YOLOv11 backbone network. For each frame , and σ (⋅) denotes non-linear activation function. Thereafter, Convolutional layers transform pixel-level information into feature representations as given in Eq. (2).

Eq. (2)

The backbone yields multi scale feature set from frame t, where in Eq. (3)

Eq. (3)

The input to the neck network is the set of multi-scale features extracted from the backbone network . First, the dimensionality of each feature map is reduced via convolutional transformations using

Eq. (4)

Next, a FFM is applied to combine features from different levels:

Eq. (5)

The fused feature map Ft​ integrates information from low-, mid-, and high-level features, providing a refined representation for the detection head.

After obtaining the fused feature maps 𝓩ₜ ​ from the neck network, the detection head is applied to predict bounding boxes, objectness scores, and class probabilities. For each frame t, the detection head generates a set of bounding box predictions . In parallel, the class probabilities are obtained by applying a softmax function to the classification logits resulting , which distributes probability across K object categories. Additionally, an Objectiveness score is produced via sigmoid function estimating the confidence that the predicted box actually contains an object.

The learning process is guided by a multi-component loss function. The localization loss ​ penalizes deviations between the predicted bounding box regression error between the predicted box and corresponding ground truth applied only to anchor positions where an object is present. This is mathematically defined as:

Eq. (6)

The objectness loss ​ evaluates the predicted objectness score​ against its ground-truth binary label ∈{0,1} ensuring the network learns to distinguish between object and background regions as in Eq. ()

Eq. (7)

The classification loss is computed only at positions containing objects, comparing the predicted class distribution Cn,j​ with the one-hot encoded ground truth across all classes. It is given by:

Eq. (8)

Finally, the total training loss combines these components into a weighted sum as in Eq. ()

Eq. (9)

Here, the trainable parameters are and optimiser update is

Frame reconstruction is performed by overlaying detected boxes, class labels onto each frame, represented as

Eq. (10)

The reconstructed video sequence is with refined using Non-Maximum Suppression.

*b. Fractal-Geometry–Driven Dynamic Watermark Embedding.*

During the imperceptible embedding process, fractal geometry employs the Sierpinski triangle’s recursive centroid distribution to imperceptibly embed watermark bits into video frame. is divided into sequence of frames as Eq. (11). Random frame is selected to embed from Dynamic watermark is generated using user attributes as Eq. (12) and video hash using Eq. (13). Final watermark is converted into binary form using Eq. (14).

Eq. (11)

where is the frame and is the total number of frames.

Eq. (12)

Eq. (13)

Eq. (14)

To define a sierpinski triangle inside frame using Eq. (15). Further, centroid of triangle is defined using Eq. (16). For recursive portioning yields centroids set use Eq. (17). To avoid the collisions among the centroid’s shifting rule is used as shown in Eq. (18).

Eq. (15)

Where *h*, *w,* are frame height and width.

Eq. (16)

Eq. (17)

Eq. (18)

Here, penultimate Least significant Bit embedding is used. For each pixel value at centroid location in grayscale using Eq. (19). Further, watermarked frame is given using Eq. (20).

Eq. (19)

Eq. (20)

Finally, watermark frame is added to remaining frames as shown in Eq. (21).

Eq. (21)

*c. Watermark Extraction and Authenticity Validation.*

From each watermark frame , the embedded dynamic watermark bits are retrieved by accessing the Penultimate least significant bit of the pixel intensities located at the centroid positions generated by the fractal geometry of sierpinski triangle. For each pixel at location with intensity value , the extracted bit is obtained as Eq. (22).

Eq. (22)

where is recovered bit from the PLSB of pixel and set of centroid coordinates generated by fractal subdivision.

Thus, the reconstructed binary watermark sequence is generated as shown in Eq. (23). Once all bits are retrieved, the sequence is grouped into 8-bit binary chunks to reconstruct the original ASCII characters. The final extracted dynamic watermark is then expressed as shown in Eq. (24).

Eq. (23)

Eq. (24)

Where maps the binary sequence back into textual form.

The authenticity of the video is determined by comparing the Extracted watermark with the that was generated during embedding. The integrity verification function is defined as shown in Eq. (25).

*Integrity(M)=*  Eq. (25)

where indicates that the video has not been tampered with and indicates possible tampering or modification. Algorithm 1 for proposed method is:

**Algorithm 1: Dynamic Video Watermarking with Fractal-Geometry Embedding**

**Input:** Video sequence , input size , Dynamic watermark   
**Output:** Watermarked video, *Integrity(M*)∈ {0,1}

1: Function Preprocess (M, d):

*′​ = R (​, d × d)*

;

Return ;

2: Function Feature\_Extraction (X):

;

,

Return

3:

4: Function Compute\_Hash(M):

Return

5: Function Generate\_Watermark

Return

6:

7: Function Embed\_Watermark (,):

;

Return ;

8: Function Extraction\_verify\_integrity (M′,,):

Return ;

9: Function main ():

)

  Return (,*I*);

**4. Results and Analysis**

To demonstrate the effectiveness of the proposed approach, Fig. 2 and Fig. 3 collectively present the process of dynamic watermark embedding and deepfake verification. As illustrated in Fig. 2, each input video is first analysed to create a dynamic watermark, which is generated by combining user attributes. Once generated, this watermark is then embedded into a carefully selected frame, thereby producing the watermarked video.

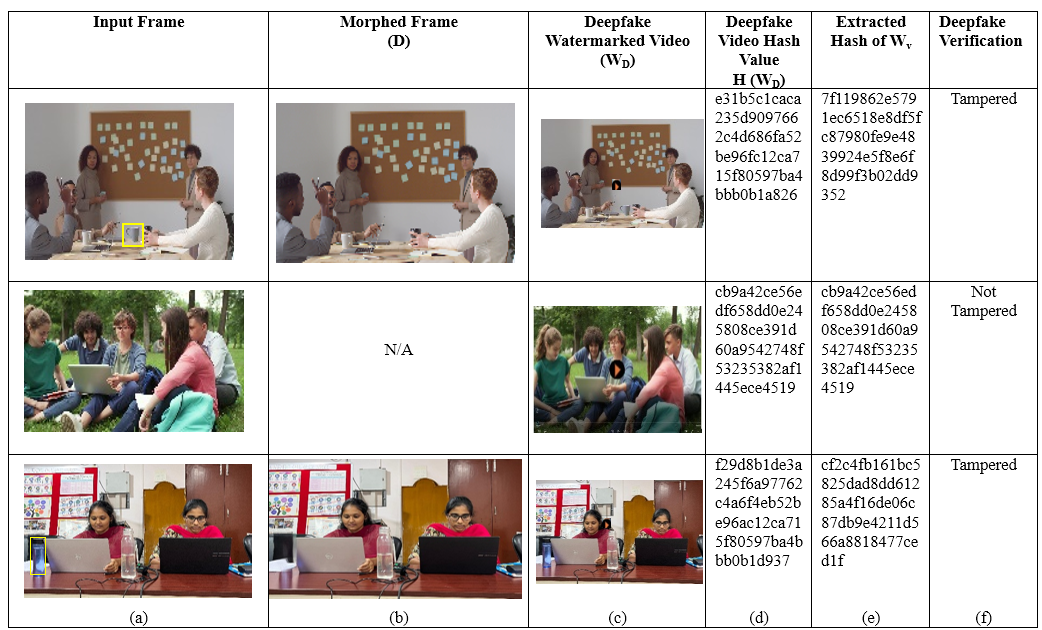
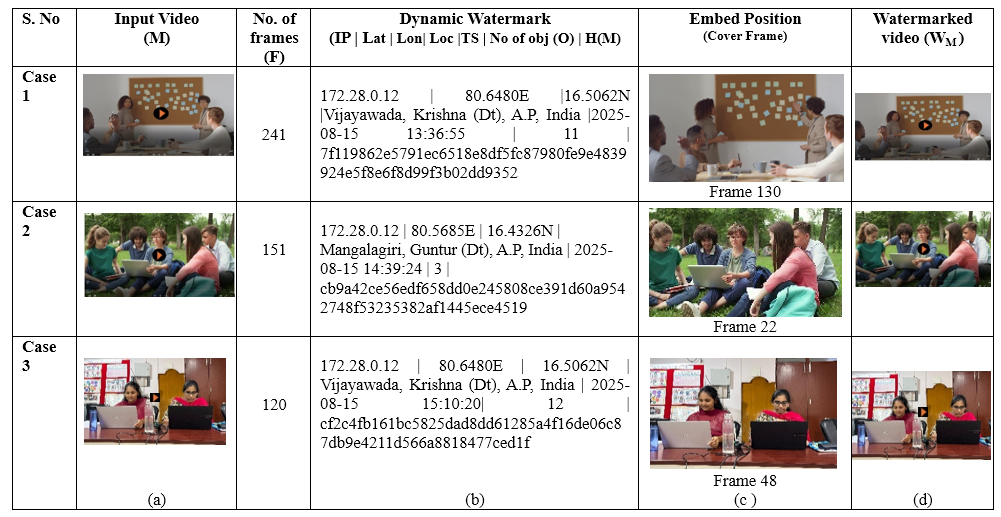
Fig 2. (a) Input video (b) Dynamic Watermark generates by considering user’s attributes like IP address (IP), latitude (Lat), Longitude (Lon), Location (Loc), Time stamp (TS) Number of Objects (O) and Original Video hash value (c) Embed Frame Position (d) Watermarked Video

Fig 3. (a) Input Frame (b) Morphed Frame (c) Deepfake Watermarked Video (d) Extracted Hash of Input Video(M) (e) Extracted has for Deepfake Video (D) (f) Deepfake Verification.

Furthermore, Fig. 3 emphasizes the detection stage, wherein input frames are systematically examined against possible morphed frames. In this stage, hash values are extracted both from the original watermarked video and the suspected deepfake video, and subsequently compared to determine authenticity, while preserving the overall integrity of multimedia content.

**5. Performance Analysis**

To comprehensively evaluate the performance of the proposed watermarking technique, several quantitative measures are utilized. These metrics assess both the visual quality of the watermarked content and the robustness of watermark extraction under various conditions. Peak Signal-to-Noise Ratio (PSNR) is employed to evaluate imperceptibility and ensure that the embedding process does not degrade visual quality with Eq. (26).

Eq. (26)

Where, L= max pixel value; = pixel value of original frame at position = watermarked frame pixel value at position ; dimensions of frame.

Bit Error Rate (BER) with Eq. (27)., Normalized Correlation (NC) Eq. (28)., and Hamming Distance (HD) are calculated using PHash to measure the accuracy and reliability of watermark recovery with Eq. (29). Furthermore, Mean Average Precision (mAP) is used to quantify the detection performance across video frames with Eq. (30), while Extraction Time evaluates the computational efficiency of the system using Eq. (31).

Eq. (27)

Where, B= total watermark bits; = kth bit in original watermark; = kth bit in extracted watermark.

Eq. (28)

Eq. (29)

Where, =1 if , else 0. B= length of watermark sequence.

Eq. (30)

Where, classes; = recall levels for class.

Eq. (31)

Where, F= total frames processed; = starting time of extraction; = ending time of extraction for frame f.

Table 3 provides a summary of the evaluation metrics employed to assess the proposed watermarking technique for mentioned three cases in Results. Fig. 4 gives the graph for the performance analysis of mentioned testcases in the Fig. 2.

Table 3. Performance analysis of different testcases

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | HD | PSNR | BER | NC |  | ET |
| Case1 | 3.777 | 48.18 | 0.0078 | 0.94 | 0.82 | 1.01 |
| Case2 | 0.32 | 47.22 | 0.0025 | 0.99 | 0.98 | 4.02 |
| Case3 | 2.32 | 51.68 | 0.0094 | 0.95 | 0.76 | 3.36 |

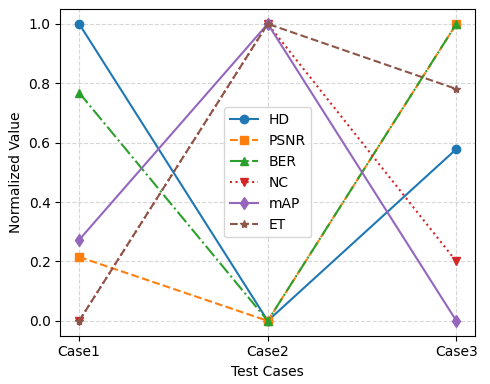


Fig. 4 performance analysis of different testcases

**6. Conclusion**

**References**

1. R. Ramanaharan, D. B. Guruge, and J. I. Agbinya, "DeepFake video detection: Insights into model generalisation — A systematic review," Data Inf. Manag, vol. 2025, p. 100099, 2025, doi: 10.1016/j.dim.2025.100099.
2. Aberna, P., Agilandeeswari, L. Digital image and video watermarking: methodologies, attacks, applications, and future directions. *Multimed Tools Appl* **83**, 5531–5591 (2024). https://doi.org/10.1007/s11042-023-15806-y
3. A. Qureshi, D. Megías and M. Kuribayashi, "Detecting Deepfake Videos using Digital Watermarking," **2021** Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Tokyo, Japan, 2021, pp. 1786-1793.
4. R. Munir and Harlili, "A Secure Fragile Video Watermarking Algorithm for Content Authentication Based on Arnold Cat Map," *2019 4th International Conference on Information Technology (InCIT)*, Bangkok, Thailand, 2019, pp. 32-37, doi: 10.1109/INCIT.2019.8912074.
5. Aouthu, S., Kollati, M., Kuraparthi, S., Mittal, A., & Joshi, A. (2024). Robust video watermarking with complex conjugate symmetric sequency transform using SURF feature matching. *Cogent Engineering*, *11*(1). https://doi.org/10.1080/23311916.2024.2345851
6. Yang, X.; Zhang, Z.; Jiao, Y.; Li, Z. A Robust Video Watermarking Algorithm Based on Two-Dimensional Discrete Fourier Transform. *Electronics* **2023**, *12*, 3271. <https://doi.org/10.3390/electronics12153271>.
7. Singh, R., Mittal, H. & Pal, R. Optimal keyframe selection-based lossless video-watermarking technique using IGSA in LWT domain for copyright protection. *Complex Intell. Syst.* **8**, 1047–1070 (2022). <https://doi.org/10.1007/s40747-021-00569-6>
8. T. Jayamalar and N. Krishnaveni, "Medical Image Video Watermarking Using Deep Neural Network Technique," 2024 Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE), Vellore, India, 2024, pp. 1-7, doi: 10.1109/ic-ETITE58242.2024.10493610.
9. L.-Y. Hsu, "AI-assisted deepfake detection using adaptive blind image watermarking," J. Vis. Commun. Image Represent., vol. 100, p. 104094, 2024, doi: 10.1016/j.jvcir.2024.104094.
10. S. J. Amruth, V. S. Shruthik and S. Jayan, "Secure Data Embedding using Fourier Transform-based Watermarking," *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Kamand, India, 2024, pp. 1-7, doi: 10.1109/ICCCNT61001.2024.10726146.
11. J. Sang, Q. Liu, and C.-L. Song, “Robust video watermarking using a hybrid DCT-DWT approach,” Journal of Electronic Science and Technology, vol. 18, no. 2, p. 100052, 2020, doi: 10.1016/j.jnlest.2020.100052.
12. M. D. Swanson, M. Kobayashi, and A. H. Tewfik, “Multimedia dataembedding and watermarking technologies,” Proceedings of the IEEE, vol. 86, no. 6, pp. 1064–1087, Jun. 1998.
13. He, Luhao & Zhou, Yongzhang & Liu, Lei & Ma, Jianhua. (2024). Research and Application of YOLOv11-Based Object Segmentation in Intelligent Recognition at Construction Sites. Buildings. 14. 3777. 10.3390/buildings14123777.
14. He, Lh., Zhou, Yz., Liu, L. *et al.* Research on object detection and recognition in remote sensing images based on YOLOv11. *Sci Rep* **15**, 14032 (2025). https://doi.org/10.1038/s41598-025-96314-x
15. Y. Sun, J. Wang, H. Huang, and Q. Chen, “Research on scalable video watermarking algorithm based on H.264 compressed domain,” *Optik*, vol. 227, p. 165911, 2021, doi: 10.1016/j.ijleo.2020.165911.
16. M. Bansal, W.Q. Yan and M. S. Kankanhalli, "Dynamic watermarking of images," *Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint*, Singapore, 2003, pp. 965-969 vol.2, doi: 10.1109/ICICS.2003.1292601.
17. Vrdoljak, Anton & Miletić, Kristina. (2019). PRINCIPLES OF FRACTAL GEOMETRY AND APPLICATIONS IN ARCHITECTURE AND CIVIL ENGINEERING. E-Zbornik elektronički zbornik radova Građevinskog fakulteta. 9. 40-52.
18. M. Holliman, W. Macy, and M. M. Yeung, “Robust frame-dependent video watermarking,” Proc. SPIE Security Watermarking Multimedia Contents II, vol. 3971, pp. 186–197, 2000.
19. Omali, Thomas. (2023). Fractal Geometry and its Application in Geographic Information Science. 9. 116-120.
20. D. Choi, H. Do, H. Choi, and T. Kim, “A blind MPEG-2 video watermarking robust to camcorder recording,” Signal Processing, vol. 90, no. 4, pp. 1327–1332, Apr. 2010.
21. Rupa, Ch. (2013). A Digital Image Steganography using Sierpinski Gasket Fractal and PLSB. Journal of The Institution of Engineers (India): Series B. 94. 147-151. 10.1007/s40031-013-0054-z.
22. T. Dutta and H. P. Gupta, “A robust watermarking framework for high efficiency video coding (HEVC) encoded video with blind extraction process,” Journal of Visual Communication and Image Representation, vol. 38, pp. 29 – 44, 2016.
23. Suchita Sharma, Shivendra Shivani, and Nitin Saxena, “An efficient fragile watermarking scheme for tamper localization in satellite images,” *Computers and Electrical Engineering*, vol. 109, Part B, 2023, Art. no. 108783, ISSN 0045-7906. doi:10.1016/j.compeleceng.2023.108783.
24. P. Samanta and S. Jain, “Analysis of Perceptual Hashing Algorithms in Image Manipulation Detection,” *Procedia Computer Science*, vol. 185, pp. 203–212, Jun. 2021, doi: 10.1016/j.procs.2021.05.021.
25. Khalaf, T.A., Mohammed, H. Bit error rate performance analysis of AC-MAP in multiple input single output wireless relay network. *J Wireless Com Network* **2020**, 12 (2020). <https://doi.org/10.1186/s13638-019-1633-8>.