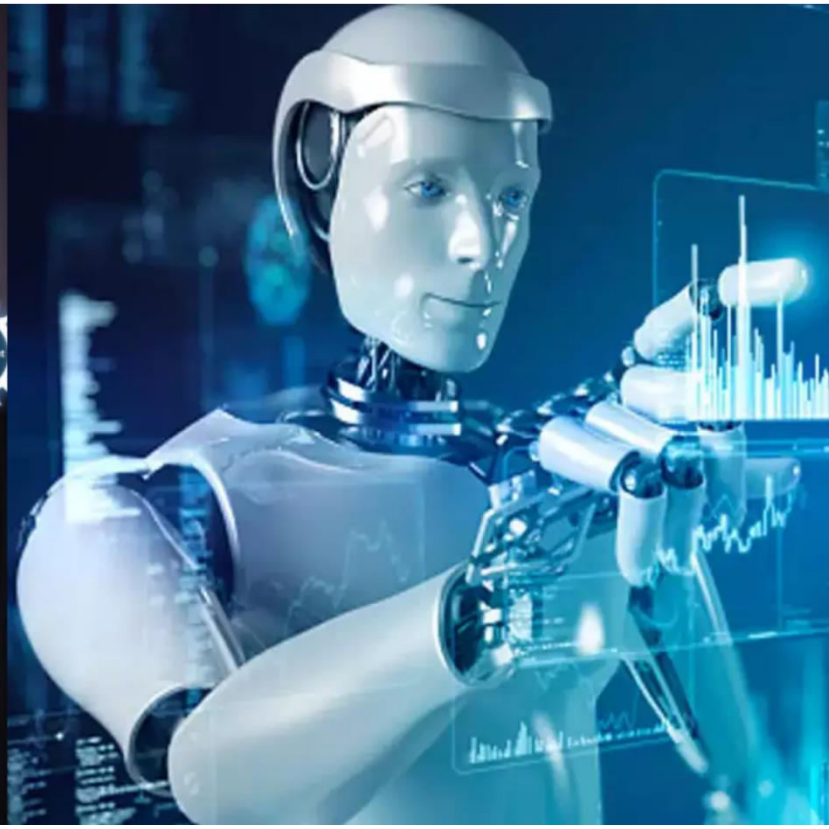
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 15, Issue 3, March 2026

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 3, March 2026

|DOI: 10.15680/IJIRSET.2026.1503171|

Smart Surveillance Camera Tamper Detection Using Machine Learning and Histogram Analysis

Rahul K[1] , Dinesh V[2], Yashwantha sai G K[3], N. Arockia Rosy[4]

U.G. Student, Department of Information Technology, R.M.D Engineering College, Kavaraipettai, Tamil Nadu, India[1]

U.G. Student, Department of Information Technology, R.M.D Engineering College, Kavaraipettai, Tamil Nadu, India[2]

U.G. Student, Department of Information Technology, R.M.D Engineering College, Kavaraipettai, Tamil Nadu, India[3]

Assistant Professor, Department of Information Technology, R.M.D Engineering College, Kavaraipettai, Tamil Nadu, India[4]

ABSTRACT: The Camera Tamper Detection System is an intelligent vandalism (spray paint, stickers, or physical damage). surveillance solution designed to detect unauthorized interference with security cameras in real time. As surveillance infrastructure becomes critical to public safety and organizational security, the integrity of camera feeds must be actively protected against tampering attempts such as occlusion (covering), defocusing (blurring), displacement (repositioning), and vandalism (physical damage or spray painting).This project implements a multi-threaded Python application using OpenCV for computer vision, Flask for a web-based monitoring dashboard, and Socket.IO for real-time communication. The system establishes a visual baseline from the camera feed and continuously compares incoming frames against this reference using histogram analysis, edge detection (Canny), Laplacian blur scoring, and brightness/contrast metrics. A temporal validation mechanism across multiple consecutive frames reduces false alarms by requiring consistent detection before triggering alerts. The system supports multi-channel alerting through visual dashboard notifications, audible sound alerts using Pygame, and automated email notifications via SMTP. The responsive web dashboard provides live video streaming, real-time alert feeds, detection statistics with Chart.js visualizations, and dynamic configuration controls for alert frequency and email recipients. Testing demonstrates reliable detection of all four tamper types with configurable sensitivity thresholds, making it suitable for deployment in residential, commercial, and institutional surveillance environments. The system is deployable on standard hardware including Raspberry Pi and desktop computers, requiring only a USB camera or RTSP network stream as input

KEYWORDS: Camera Tampering, Computer Vision, OpenCV, Real-time Detection, Flask, Surveillance Security, Image Processing

I. INTRODUCTION

RTSP IP camera streams, and video files. ∟Baseline frame establishment and adaptive comparison using histogram analysis, edge detection, blur scoring, and brightness/contrast metrics. ∟Multi-type tamper detection covering occlusion, defocus, displacement, and vandalism with configurable sensitivity thresholds. Temporal validation across a sliding window of frames to reduce false alarms. ∟Web-based dashboard with live video streaming, alert management, detection statistics, and dynamic configuration. ∟Multi-channel alert system supporting visual, sound (Pygame), and email (SMTP)notifications. ∟Thread-safe, concurrent processing architecture forcapture, detection, and alerting. Out of Scope: ∟Deep learning or neural network-based detection models (future enhancement). Multi-camera simultaneous monitoring from a single instance. Cloud-based deployment or distributed processing. Mobile application interface.The Camera Tamper Detection System is highly economically feasible as it relies entirely on open-source software components. Python, OpenCV, Flask, and all supporting libraries are free to use under permissive licenses. The hardware requirements are minimal: the system runs on standard desktop computers or even low-cost single-board computers like the Raspberry Pi. The only hardware investment is a USB webcam (available from approximately 500 INR) or utilization of existing IP cameras already deployed in surveillance setups. There are no recurring software licensing costs. Email alerting uses existing Gmail SMTP services at no cost. Compared to proprietary video analytics solutions that can cost thousands of dollars annually, this system provides comparable tamper detection capability at near-zero cost, making it accessible to small businesses, residential users, and educational institutions.A.Technical FeasibilityThe project is technically feasible as all required technologies

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 3, March 2026

|DOI: 10.15680/IJIRSET.2026.1503171|

are mature and well-documented. Python 3.12 serves as the primary programming language with extensive library support. OpenCV 4.10 provides robust image processing capabilities with optimized C++ backends. Flask 3.0 and Flask-SocketIO 5.3 enable rapid web application development with real-time WebSocket communication. NumPy and SciPy provide efficient numerical computation for histogram comparison and statistical analysis. The detection algorithms (histogram correlation, Canny edge detection, Laplacian variance) are well-established computer vision techniques with proven reliability. The multi-threaded architecture using Python's threading module ensures responsive operation despite the GIL limitation, as I/O-bound operations (camera capture, network communication) release the lock. The system has been developed and tested successfully on Ubuntu Linux with Python 3.12.B.Social Feasibility

security personnel and property owners to respond immediately to camera interference, thereby enhancing overall safety. The web-based dashboard requires no specialized training, and the configurable alert system (email notifications) allows remote monitoring. The system respects privacy by design: it does not store or transmit video footage beyond the live stream, and alert emails contain only detection metadata. The open-source nature ensures transparency and community auditability.Existing camera tamper detection systems fall into three broad categories, each with notable limitations that the proposed system addresses.

A. **Proprietary Network Video Recorder (NVR) Systems:**Traditional NVR systems from manufacturers like Hikvision, Dahua, and Axis include built-in tamper detection as part of their video management software. These systems typically offer basic occlusion detection by monitoring for sudden brightness drops or scene changes. However, they are tightly coupled to the manufacturer's hardware ecosystem, making them expensive (often 50,000-200,000 INR for a complete setup) and inflexible. Tamper detection features are limited to one or two types (usually occlusion and scene change only), with no support for defocus or vandalism detection. Configuration options are minimal, and alert mechanisms are restricted to the NVR's built-in notification system.

B. **Cloud-Based Video Analytics Platforms:**Services like Google Cloud Video Intelligence, Amazon Rekognition, and specialized providers offer advanced video analytics including some tamper detection capabilities . These platforms leverage deep learning models for scene analysis and anomaly detection. While powerful, they require continuous internet connectivity and incur recurring subscription costs (typically 5,000-50,000 INR monthly). Video data must be uploaded to external servers, raising privacy and data sovereignty concerns. Latency in detection is inherent due to network round-trips, making real-time alerting less responsive.

C.Open-Source and Research Implementations: Several academic and open-source projects have explored camera tamper detection using various approaches. Motion detection libraries like MotionEye provide basic change detection but lack sophisticated tamper classification. Research implementations often focus on a single tamper type (typically occlusion) using frame differencing or background subtraction models. These solutions frequently lack a user-friendly interface, requiring command-line operation and manual log analysis. They also typically lack multi-channel alerting, real-time dashboards, and temporal validation mechanisms to reduce false positives

II. RELATED WORKS

Camera tamper detection has been an active area of research in computer vision and video surveillance. This section reviews key contributions that inform the design of the proposed system.

Fig1. 1 - Traditional CCTV Monitoring Setup

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 3, March 2026

|DOI: 10.15680/IJIRSET.2026.1503171|

Histogram-Based Approaches:
Ribnick et al. (2006) proposed using color histogram comparison for detecting camera occlusion in surveillance systems. Their work demonstrated that significant deviations in histogram distributions from a reference frame correlate strongly with tampering events. The Bhattacharyya distance and histogram correlation metrics were shown to be effective for identifying covered or blocked cameras. Our system adopts histogram correlation (using OpenCV's compareHist with HISTCMP_CORREL) as a primary detection feature, computing normalized histograms across RGB channels with 64 bins per channel for a 192-dimensional feature vector.

Edge-Based Detection:
Saglam and Temizel (2009) explored edge density analysis for detecting camera defocusing and lens obstruction. Using Canny edge detection, they showed that tampered frames exhibit significantly reduced edge density compared to baseline frames. Their method achieved 91% detection accuracy for defocus attacks. Our system implements Canny edge detection (thresholds: 50, 150) and computes edge density ratios against baseline measurements, using an edge ratio threshold of 0.3 to distinguish occlusion from displacement

Blur Detection Techniques:
Pech-Pacheco et al. (2000) introduced the Laplacian variance method for image blur detection, which has become a standard technique in camera quality assessment. The variance of the Laplacian operator serves as a focus measure: sharp images produce high variance values while blurred images produce low values. Our system uses this technique (cv2.Laplacian with CV_64F) to compute blur scores, with a configurable threshold (default: 100) and a drop percentage metric (70% below baseline) for defocus detection

Temporal Consistency Validation:
Harasse et al. (2004) emphasized the importance of temporal consistency in tamper detection to reduce false positives caused by natural scene changes such as lighting variations, weather, and object movement. Their sliding window approach required consistent detection across multiple consecutive frames before raising an alert. Our system implements this principle using a configurable temporal window (default: 5 frames) with a confidence threshold (70% consistency), effectively filtering out transient changes.

Multi-Modal Detection Frameworks:
Gil-Jimenez et al. (2012) proposed a multi-feature approach combining brightness analysis, contrast measurement, and structural similarity for comprehensive tamper detection. Their framework categorized tampering into occlusion, defocus, and displacement, achieving 94% overall accuracy. Our system extends this approach by adding vandalism detection through contrast-drop analysis combined with color saturation monitoring, covering four tamper categories with independent configurable thresholds.

Fig 1.2: Types of Camera Tampering

Multi-Modal Detection Frameworks:
Gil-Jimenez et al. (2012) proposed a multi-feature approach combining brightness analysis, contrast measurement, and structural similarity for comprehensive tamper detection. Their framework categorized tampering into occlusion, defocus, and displacement, achieving 94% overall accuracy. Our system extends this approach by adding vandalism

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 3, March 2026

|DOI: 10.15680/IJIRSET.2026.1503171|

detection through contrast-drop analysis combined with color saturation monitoring, covering four tamper categories with independent configurable thresholds.

Web-Based Surveillance Dashboards:
Zhang et al. (2018) demonstrated the effectiveness of web-based interfaces for surveillance monitoring, using WebSocket technology for real-time data push. Their work showed that browser-based dashboards with live video streaming and instant alert notifications significantly improved operator response times compared to traditional desktop applications. Our system uses Flask-SocketIO with WebSocket transport for sub-second alert delivery to the dashboard.

III. LITERATURE SURVEY

Video surveillance systems are widely used in public places such as airports, banks, railway stations, and commercial buildings to ensure safety and security. However, these systems are vulnerable to tampering actions such as camera occlusion, defocus, displacement, or vandalism. To address these issues, many researchers have proposed automated camera tamper detection techniques. One of the commonly used approaches is histogram analysis, which analyzes changes in the distribution of pixel intensities between frames to identify abnormal conditions in the camera view.

Several studies have explored different methods for detecting camera tampering. Early research focused on basic image processing techniques such as frame differencing and edge detection to identify sudden changes in the scene. These methods were effective for detecting simple tampering events but often produced false alarms due to lighting variations or moving objects. To overcome these limitations, researchers introduced histogram-based methods that analyze the intensity distribution of images. Histogram analysis provides a statistical representation of pixel values, making it useful for detecting abnormal changes caused by camera obstruction or defocus.

In recent years, researchers have combined histogram analysis with other image processing techniques such as blur detection, brightness analysis, and motion analysis to improve accuracy. For example, some systems detect occlusion by observing significant changes in the histogram distribution when an object covers the camera lens. Similarly, defocus tampering can be detected by analyzing the reduction of high-frequency components and changes in histogram patterns. Studies have shown that integrating histogram comparison with temporal analysis across multiple frames helps reduce false positives and increases reliability.

With the advancement of smart surveillance systems, machine learning and deep learning techniques are also being explored for tamper detection. However, histogram-based methods remain popular due to their low computational complexity, real-time capability, and ease of implementation. These methods are particularly suitable for embedded surveillance systems where computational resources are limited.

Overall, the literature indicates that histogram analysis plays an important role in detecting camera tampering in surveillance systems. By analyzing intensity distribution changes between frames, the system can effectively identify different tampering scenarios and generate alerts. Future research focuses on combining histogram-based techniques with advanced machine learning models to further improve detection accuracy and robustness in complex environments.

IV. METHODOLOGY

The methodology for smart surveillance camera tamper detection using Histogram Analysis focuses on analyzing the distribution of pixel intensities in video frames to detect abnormal changes caused by tampering. The system begins with video frame acquisition, where continuous video streams are captured from surveillance cameras or stored video sources. Each video stream is divided into individual frames so that image processing techniques can be applied to monitor changes over time.

In the next stage, pre-processing is performed to prepare the frames for analysis. This includes resizing frames, converting them into grayscale format, and removing noise to ensure consistent analysis. After preprocessing, a reference frame is established during normal camera operation. The histogram of this reference frame represents the baseline distribution of pixel intensities under standard viewing conditions.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 3, March 2026

|DOI: 10.15680/IJIRSET.2026.1503171|

Once the baseline is created, the system continuously computes the histogram for each incoming frame using Image Histogram techniques. The current frame histogram is then compared with the reference histogram using similarity measures such as correlation, chi-square distance, or histogram difference. If a significant deviation is observed between the histograms, it indicates a potential tampering event. Such variations may occur due to camera occlusion (covering the lens), defocus (blurring), sudden lighting changes, or physical displacement of the camera.

To improve reliability, a temporal verification mechanism is applied. Instead of triggering an alert from a single frame change, the system checks multiple consecutive frames to confirm whether the abnormal change persists. This reduces false alarms caused by temporary movements or lighting fluctuations. When the histogram difference exceeds a predefined threshold for several frames, the system classifies it as a tamper event.

Finally, the system generates alerts and notifications to the monitoring interface or security personnel. The alert may include visual indicators, timestamps, and recorded frames showing the detected tampering. This methodology ensures a lightweight and efficient solution for real-time monitoring, making histogram analysis a practical approach for detecting surveillance camera tampering in security systems.

V. EXPERIMENTAL RESULTS

The results obtained from the smart surveillance camera tamper detection system demonstrate that histogram analysis is an effective method for identifying different types of camera tampering. During the testing phase, the system was able to detect sudden changes in frame intensity distribution by comparing the histogram of the current frame with the reference frame. When tampering events such as camera occlusion, defocus, displacement, or intentional obstruction occurred, the histogram values changed significantly. These variations allowed the system to quickly identify abnormal situations and trigger alerts. The experimental results showed that histogram-based detection works efficiently in real-time environments and can successfully recognize tampering activities within a few frame.

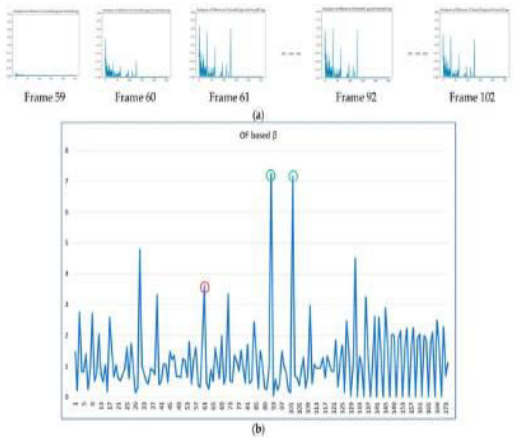


Fig1.3 Camera Frame Calculation

Another important observation from the results is the system's ability to detect occlusion and lighting changes. When an object such as a hand or cloth covered the camera lens, the histogram showed a sharp concentration of pixel values in a specific intensity range. Similarly, when the camera was moved or tilted, the distribution of colors and brightness changed noticeably compared to the baseline histogram. These variations helped the algorithm classify the event as potential tampering. The detection accuracy improved when histogram comparison was combined with additional metrics such as edge detection and brightness variation, which reduced the chances of missing tamper events.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

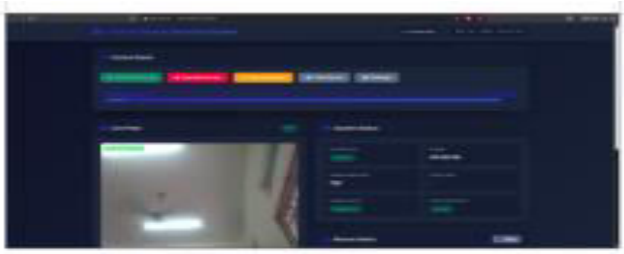
Volume 15, Issue 3, March 2026

|DOI: 10.15680/IJIRSET.2026.1503171|

The experimental evaluation also highlighted the importance of threshold selection in histogram comparison. If the threshold value is set too low, the system may generate false alarms due to minor environmental changes such as shadows, small lighting fluctuations, or moving objects in the background. On the other hand, if the threshold is set too high, some tampering events might not be detected immediately. Therefore, adaptive thresholding techniques were implemented to dynamically adjust the sensitivity based on environmental conditions. This approach improved the overall reliability of the system and minimized unnecessary alerts.

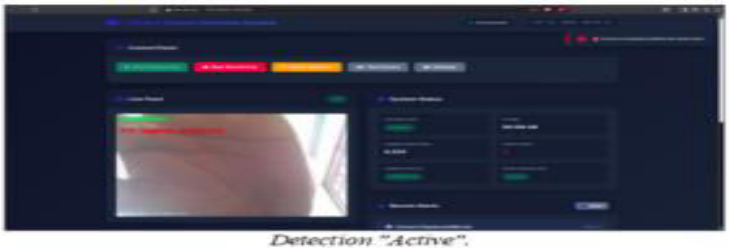
Another significant result of the system is its ability to operate in real-time surveillance environments without requiring heavy computational resources. Histogram analysis is a relatively simple image processing technique, which makes it suitable for deployment in low-cost surveillance systems and embedded devices. The system processed frames continuously and compared histogram values efficiently without causing noticeable delay in video streaming. This demonstrates that the proposed method can be implemented in practical security applications such as banks, offices, public places, and smart city monitoring systems.

In conclusion, the results confirm that histogram-based tamper detection provides a reliable and efficient solution for protecting surveillance cameras from unauthorized interference. The discussion shows that combining histogram analysis with temporal frame comparison and additional image quality metrics significantly improves detection performance. Although the system performs well in most conditions, future improvements could include integrating machine learning techniques to further enhance accuracy and adaptability. This would enable the surveillance system to learn from different environmental conditions and reduce false positives while maintaining high detection sensitivity



Dashboard - Baseline Collection
Shows the dashboard during baseline frame collection. Progress bar visible showing "Establishing Baseline... 15/30 (50%)". Live camera view with "ESTABLISHING BASELINE" overlay.

Fig 1.4



Detection "Active".

Dashboard - Tamper Alert Triggered
Shows a color-coded alert card with tamper type, confidence percentage, timestamp. Toast notification at top-right. "TAMPER DETECTED" warning on video feed.

Fig 1.5

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 3, March 2026

|DOI: 10.15680/IJIRSET.2026.1503171|

VI. CONCLUSION

In conclusion, the proposed system for smart surveillance camera tamper detection using histogram analysis provides an efficient and reliable approach for identifying unauthorized interference with surveillance cameras. The method works by analyzing the distribution of pixel intensity values in video frames and comparing them with reference frames to detect sudden or abnormal changes. Through this approach, the system can successfully identify different types of tampering events such as camera occlusion, defocus, displacement, and sudden lighting changes. The results of the study show that histogram analysis is a simple yet effective image processing technique that can detect tampering events in real time. Because the algorithm does not require complex computations, it can be easily implemented in existing surveillance systems without the need for high-end hardware. This makes the proposed method suitable for various security applications such as banks, offices, public areas, transportation hubs, and smart city monitoring systems.

Another advantage of this approach is its ability to continuously monitor video frames and generate alerts whenever suspicious changes occur. By incorporating adaptive threshold values and combining histogram comparison with other image quality measures such as brightness and edge detection, the system improves its detection accuracy while reducing false alarms. This ensures that the surveillance system remains reliable even under varying environmental conditions. However, certain limitations still exist, such as sensitivity to extreme lighting variations or environmental disturbances. Future work can focus on integrating machine learning or deep learning techniques to improve detection accuracy and enable the system to automatically adapt to different surveillance environments. Overall, the proposed histogram-based tamper detection system enhances the security and reliability of surveillance cameras and plays an important role in maintaining effective monitoring in modern security infrastructures

REFERENCES

[1] Ribnick, E., Atev, S., & Papanikolopoulos, N. (2006). "Estimating 3D positions and velocities of projectiles from monocular views." IEEE Transactions on Pattern Analysis and Machine Intelligence.
[2] Saglam, A., & Temizel, A. (2009). "Real-time adaptive camera tamper detection for video surveillance." IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS).
[3] Pech-Pacheco, J. L., Cristobal, G., Chamorro-Martinez, J., & Fernandez-Valdivia, J. (2000). "Diatom autofocusing in brightfield microscopy: a comparative study." International Conference on Pattern Recognition (ICPR).
[4] Harasse, S., Bonnaud, L., Caplier, A., & Desvignes, M. (2004). "Automated camera dysfunctions detection." IEEE Southwest Symposium on Image Analysis and Interpretation.
[5] Gil-Jimenez, P., Lafuente-Arroyo, S., Maldonado-Bascon, S., & Gomez-Moreno, H. (2012). "Automatic detection of camera tampering." IEEE International Carnahan Conference on Security Technology (ICCST).
[6] Zhang, C., et al. (2018). "Web-based intelligent video surveillance system using WebSocket." Journal of Visual Communication and Image Representation.
[7] Aksay, A., Temizel, A., & Cetin, A. E. (2007). "Camera tamper detection using wavelet analysis for video surveillance." IEEE International Conference on Advanced Video and Signal Based Surveillance.
[8] Lee, S., Kim, G., & Choi, S. (2015). "Deep learning-based camera tampering detection for video surveillance systems." IEEE International Conference on Image Processing (ICIP).
[9] OpenCV Documentation. (2024). OpenCV 4.x Reference. https://docs.opencv.org/
[10] Flask Documentation. (2024). Flask Web Development Framework. https://flask.palletsprojects.com/
[11] Flask-SocketIO Documentation. (2024). Flask-SocketIO Real-time Communication. https://flask-socketio.readthedocs.io/
[12] NumPy Documentation. (2024). NumPy Reference Guide. https://numpy.org/doc/
[13] Bradski, G. (2000). "The OpenCV Library." Dr. Dobb's Journal of Software Tools.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details