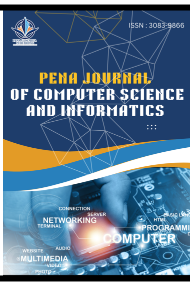
PENA CENDEKIA PUBLISHING
NURTURING IDEAS, ENRICHING KNOWLEDGE

Pena Journal of Computer Science and Informatics

Journal homepage:
https://penacendekia.com.my/index.php/pjcsi/index
ISSN: 3083-9866



Application of Hybrid Framework using Gauss Jordan Elimination, Singular Value Decomposition and Linear Binary Pattern Histogram for Image Tampering Recovery

Nurul Ain Che Intan[1], Suhaila Abd Halim[1,2,*], Siti Salmah Yasiran[1], Normi Abdul Hadi[1], Syamsudhuha[3]

[1] Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia
[2] Smart Manufacturing Research Institute, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia
[3] Faculty of Mathematics, Universitas Riau, Pekan baru 28293, Indonesia

ARTICLE INFO

Article history:
Received 5 July 2025
Received in revised form 31 August 2025
Accepted 8 September 2025
Available online 18 September 2025

Keywords:
Image tampering detection; Gauss Jordan Elimination; singular value decomposition; local binary pattern histogram

ABSTRACT

The progress in the last few years in digital editing tools has made image tampering easier. This poses a serious concern in digital forensics, journalism, and law enforcement because manipulated digital content can mislead investigations, spread misinformation, and compromise the integrity of evidence. The research proposes a hybrid framework for image tampering detection and recovery based on Gauss Jordan Elimination (GJE), Singular Value Decomposition (SVD), and Linear Binary Pattern (LBP) Histogram methods. The main goals are to detect the tampered areas accurately, determine the forgery type, which is mainly copy-move and splicing forgery, and recover the original image with minimal distortion. The CASIA 2.0 dataset was used for training and testing. Preprocessing steps included image grayscale conversion and normalization to standardize inputs. Feature extraction involved LBP histogram- based feature extraction while SVD was used for capturing. structural changes and pixel recovery was performed by GJE. The performance of the model was evaluated using PoDA, accuracy, precision, recall, F1-score, memory usage, PSNR, and MSE. On average, results show that the detection of the proposed method gives satisfactory results after copy move forgery as compared with splicing forgery. The research provides a robust and detailed framework for tampering detection and recovery of images in real time and classified images requiring precise and swift digital image verification.

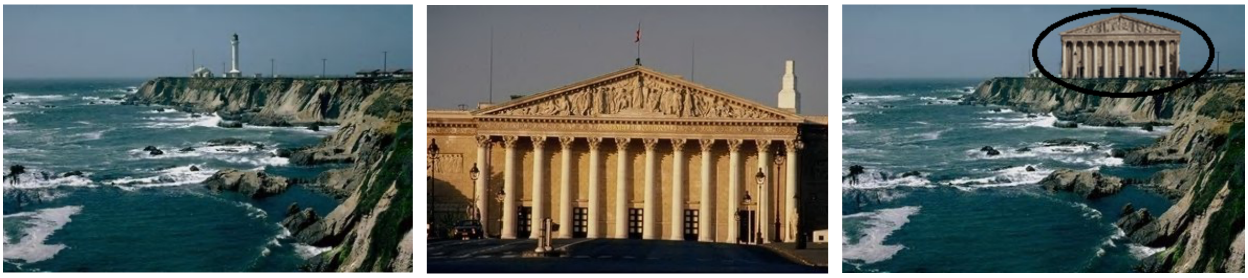
1. Introduction

The detection and recovery of tampered images have emerged as critical research areas in digital forensics, security, and content authenticity. With the increasing accessibility of advanced editing

[*] *Corresponding author.*
E-mail address: suhaila889@uitm.edu.my

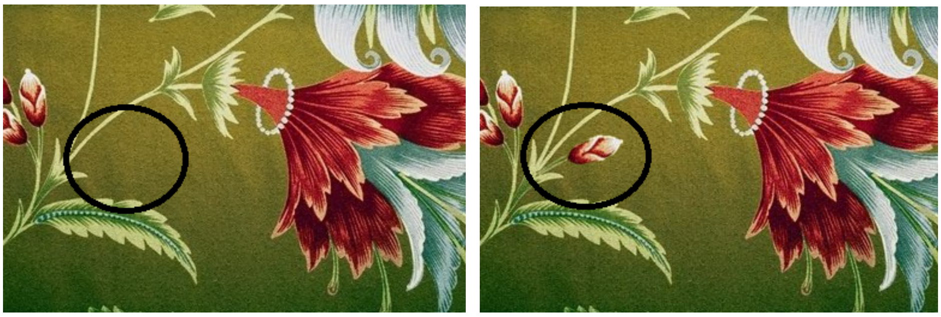
https://doi.org/10.37934/pjcsi.2.1.110

tools such as Adobe Photoshop and AI-based image generators, manipulation techniques like splicing and Copy Move Forgery as in Figure 1 and Figure 2 respectively.



(a) Source image (b) Tampered image (c) Splicing forgery

Fig. 1. Example of Splicing Forgery (a) Source image (b) Tampered image and (c) Copy move forgery



(a) Source image (b) Tampered image

Fig. 2. Example of Copy Move Forgery (a) source image (b) tampered image

Image manipulation using software has become more sophisticated and harder to identify by the human eye [1-6]. Such manipulations not only undermine trust in digital media but also pose risks in sensitive domains, including journalism, legal evidence, and national security [7]. Consequently, there is a growing need for reliable frameworks capable of detecting and recovering manipulated regions with high accuracy, robustness, and computational efficiency.

Although significant progress has been made in image tampering detection, current methods still face notable challenges. Existing approaches often fail when applied to low-quality or compressed images, are sensitive to post-processing operations such as rotation and scaling and lack robustness against both splicing and copy-move forgeries [4-6,8-11]. Moreover, many studies have focused on detection alone, with limited attention given to recovery, which is crucial for restoring the authenticity of tampered images. These limitations highlight the necessity of developing integrated solutions that combine multiple mathematical and computational approaches to improve detection precision, recovery accuracy, and adaptability to real-world conditions.

The research addresses these challenges by proposing a unified framework that integrates GJE, SVD, and LBP histogram. Each method contributes complementary strengths: GJE is effective for pixel-level self-recovery operations [12], SVD provides robust performance in feature extraction and structural analysis [9], and LBP enables efficient texture-based anomaly detection [7]. While earlier studies have applied these methods individually or in pairs, few have attempted to combine all three within a single framework, particularly with a focus on both detection and recovery using benchmark datasets such as CASIA 2.0. The integration of these methods enables the framework to capture statistical, structural, and textural inconsistencies simultaneously, thereby achieving stronger performance than standalone techniques.

The significance of this study lies in its contribution toward a practical and mathematically interpretable solution for image tampering detection and recovery. By validating the framework using the CASIA 2.0 dataset [13] and evaluating its performance through metrics such as accuracy, precision, recall, F1-score, PSNR, MSE, and memory, this research aims to provide a balanced approach that addresses accuracy, robustness, and efficiency.

The aim of the study is detecting the on tampered image. which contribute to the development of a framework for detecting and recovering tampered images by integrating GJE, SVD, and LBP histogram on tampered images from the CASIA 2.0 dataset and evaluation of the performance of the proposed framework using accuracy, computational efficiency, and robustness.

2. Methodology

The proposed framework integrates GJE, SVD, and LBP histogram for detecting and recovering tampered regions in images. The process involves five main stages: data collection, preprocessing, feature extraction, data analysis, and performance evaluation.

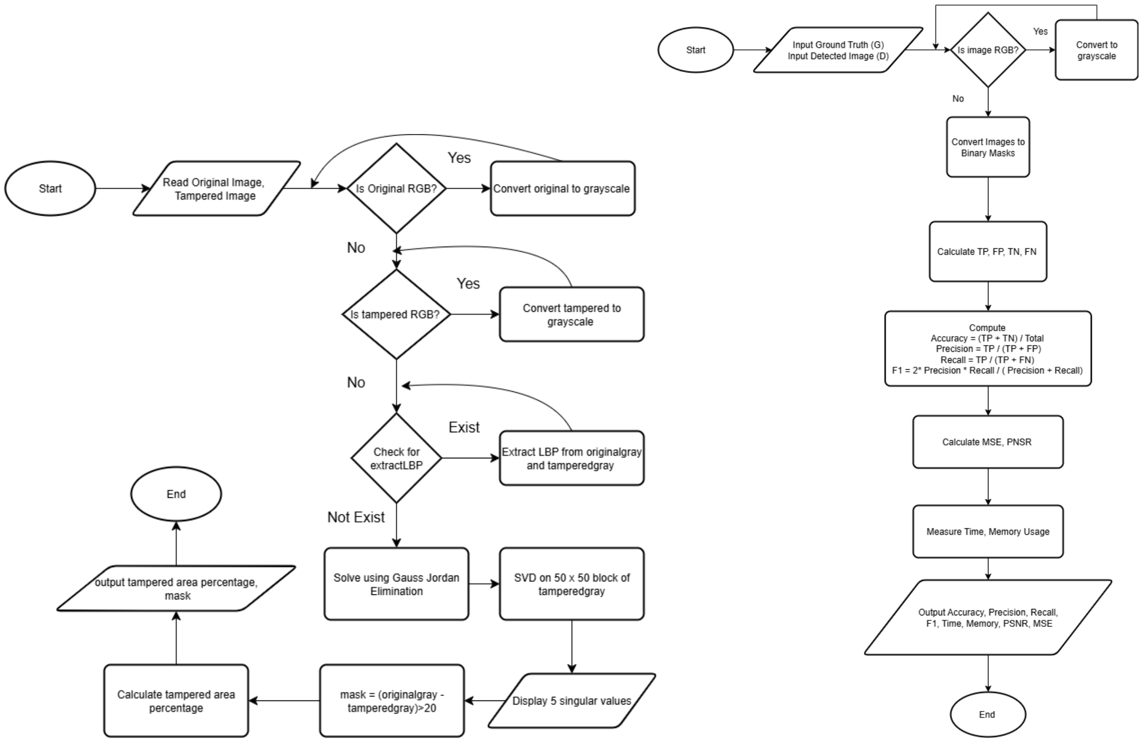

Start
Read Original Image, Tampered Image
Is Original RGB?
Yes
Convert original to grayscale
No
Is tampered RGB?
Yes
Convert tampered to grayscale
No
Check for extractLBP
Exist
Extract LBP from originalgray and tamperedgray
Not Exist
Solve using Gauss Jordan Elimination
SVD on 50 x 50 block of tamperedgray
Display 5 singular values
mask = (originalgray - tamperedgray)>20
Calculate tampered area percentage
output tampered area percentage, mask
End
Start
Input Ground Truth (G)
Input Detected Image (D)
Is image RGB?
Yes
Convert to grayscale
No
Convert Images to Binary Masks
Calculate TP, FP, TN, FN
Compute
Accuracy = (TP + TN) / Total
Precision = TP / (TP + FP)
Recall = TP / (TP + FN)
F1 = 2* Precision * Recall / (Precision + Recall)
Calculate MSE, PNSR
Measure Time, Memory Usage
Output Accuracy, Precision, Recall, F1, Time, Memory, PSNR, MSE
End

Fig. 3. Research flow of the proposed algorithm

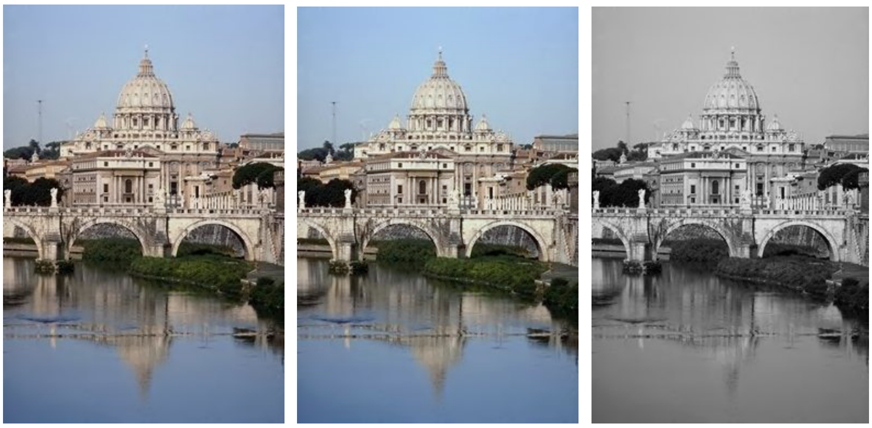
2.1 Data Collection

This study uses the CASIA 2.0 Image Tampering Detection Dataset, sourced from a public digital forensics' repository on Kaggle. The dataset contains authentic and manipulated images, primarily involving splicing and copy move tampering, which are essential for evaluating the framework's detection and recovery capabilities [14]. All images are in JPEG format with varying resolutions, enabling robustness testing under different conditions. Ground truth masks are available for certain images to aid in validation. For this study, 5 tampered images were selected for experimental implementation.

2.2 Preprocessing

Preprocessing was performed to standardize the images and enhance the accuracy of feature extraction. All images were resized to 256×256 pixels and normalized to pixel intensity range of [0,1]. Conversion to grayscale was applied to reduce computational complexity while preserving essential structural information. For evaluation purposes, region masking was performed using binary ground truth masks to isolate manipulated areas. These preprocessing steps ensured consistent input quality and improved the reliability of subsequent detection and recovery processes.

Initially the process starts with grayscale conversion and normalization aim for standardizing pixel intensity. Figure 4 shows the result of an image to be converted to grayscale image.



(a) Tampered image (b) Normalization (c) Grayscale

Fig. 4. Result of grayscale for the tampered Image

2.3 Feature Extraction

The proposed framework employs three complementary techniques for feature extraction: HJE, SVD, and LBP Histogram.

Gauss Jordan Elimination

GJE is used to identify and reconstruct tampered pixel blocks by solving a system of linear equations:

$$A\,.X\ =\ B \tag{1}$$

where A is a coefficient matrix, B represents the manipulated image values, and X is the unknown vector. The solution is obtained through:

$$X\ =\ A^{-1}.B \tag{2}$$

where A^{-1} is the inverse of A [12]. This method ensures precise pixel recovery in tampered regions.

Singular Value Decomposition

SVD decomposes an image matrix I into three components:

$$I\ =\ U\,.\sum_{\square}^{\square}\square\ .V^{T} \tag{3}$$

where U and V^T are orthogonal matrices and summation contains the singular values [9]. Tampered areas exhibit inconsistencies in singular values, aiding in manipulation detection.

Linear Binary Pattern

LBP captures local texture variations between tampered and non-tampered regions by thresholding the neighborhood of each pixel and encoding the result into a binary pattern [15]. The LBP descriptor is calculated as:

$$LBP_{(p,r)} = \sum_{j=1}^{P-1} S(N_j - N_c) \,.\, 2^j \tag{4}$$

where N_j is the intensity of the j-th neighbor, N_c is the intensity of the central pixel and $S(x) = 1$ if $x \geq 0$ else 0. Histograms of LBP values are then used as classification features for tampering detection.

2.4 Data Analysis

Data analysis involved evaluating the performance of the GJE, SVD, and LBP methods in detecting tampered regions and supporting image restoration. Detection accuracy was quantified by using standard metrics such as True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN). Computational efficiency was assessed through execution time and memory usage, while robustness was examined by testing against noise and image distortions. The quality of reconstructed images was measured using Peak Signal-to-Noise Ratio (PSNR). All analyses were conducted using MATLAB R2023a.

2.5 Performance Evaluation

The performance of the proposed GJE, SVD, and LBP-based framework was assessed in terms of accuracy, computational efficiency, and robustness. The evaluation process compared the detected tampered regions against ground truth masks, with performance metrics computed based on True Positive (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN).

Accuracy was calculated using:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision measured the proportion of correctly detected tampered regions:

$$Precision = \frac{TP}{TP + FP}$$

Recall quantified the proportion of actual tampered regions correctly identified:

$$Recall = \frac{TP}{TP + FN}$$

The F1 - score was then derived as :

$$F1 = \frac{2(Precision)(Recall)}{Precision + Recall}$$

Computational efficiency was evaluated based on memory usage, where calculated as:

$$Memory\ Usage = Size\ of\ Image \times Data\ Type\ Size$$

Robustness was measured using Peak Signal-to-Noise Ratio (PSNR) between the recovered and original images.

$$PSNR = 10 \, . log_{10}(\frac{MAX_I^2}{MSE})$$

where MAX_I is the maximum pixel value, and the Mean Squared Error (MSE) is given by:

$$MSE = \frac{1}{m \, . n} \sum_{i=1}^{m} \sum_{j=1}^{n} (I(i,j) - K(i,j))^2$$

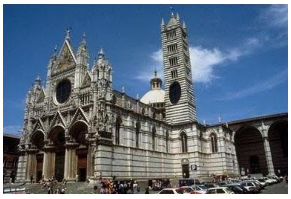
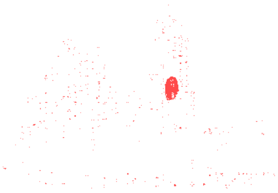
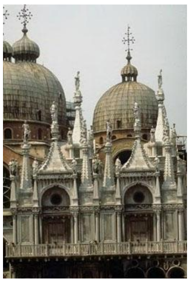
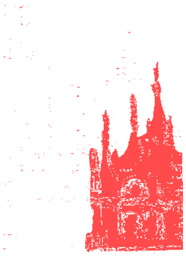
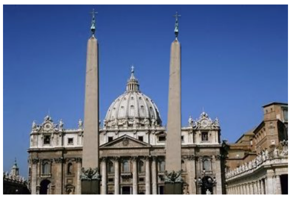
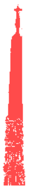
Here $I(i,j)$ and $K(i,j)$ represent the pixel intensities of the original and reconstructed images, respectively. A higher PSNR value indicates superior reconstruction quality and greater robustness against noise and distortions.

3. Result

3.1 Tampering Detection Copy Move Forgery

Table 1 demonstrates 5 images used in the tampering detection algorithm. For each image, the table presents original images, tampered images, the detected areas, and the percentages of the detected areas (PoDA). The PoDA is the percentage of a portion of detected area over the whole objects This offers a comparison of the original and tampered images side by side, giving a greater understanding of how well the system detects and measures the tampering portions.

Table 1
Copy Move Forgery Detection for five images

Bil.	Original Image	Tampered Image	Detected Areas	PoDA(%)
1.				1.01%
2.				20.70%
3.				4.30%

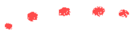
4.				1.19%
5.				0.60%

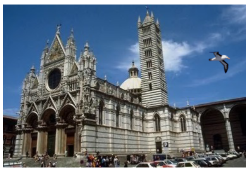
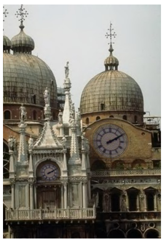
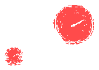
Table 1 shows that the percentage of detected areas (PoDA) for copy-move forgery varies widely across the five images. Detection ranges from as low as 0.60% to as high as 20.70%, indicating that the method can identify even small, tampered regions, though the extent of detection depends on the complexity of the forgery in each image.

3.2 *Tampering Detection Spicing Forgery*

Splicing Forgery

Table 2 demonstrates the results of 5 images used in the splicing tampered detection algorithm. This offers a comparison of both images to get the detected areas as follows:

Table 2
Splicing Forgery’s Detection for five images

Bil.	Original Image	Tampered Image	Detected Areas	PoDA (%)
1.				0.44%
2.				3.60%

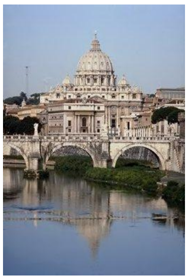
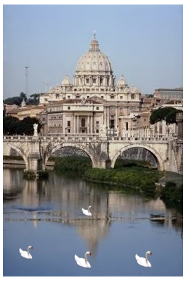
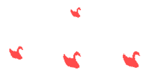
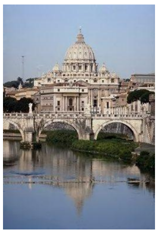
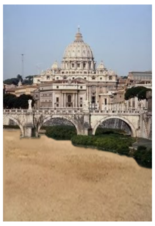
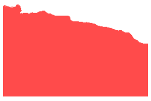
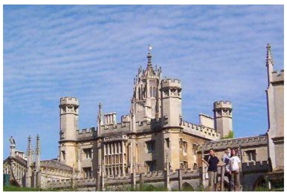
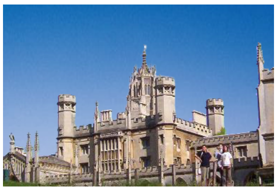
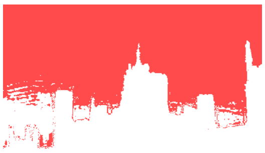
3.				0.98%
4.				35.22%
5.				51.95%

Table 2 shows that splicing forgery detection varies greatly depending on the image. The detected areas range from very small, which are 0.44% to 0.98%, to large, tampered regions, which are 35.22% to 51.95%. This indicates the method can detect both minor and extensive splicing manipulations effectively.

3.3 Performance Evaluation of Tampering Detection

Table 3 demonstrates evaluation results which are PoDA, accuracy, precision, recall, F1, PSNR, and MSE for copy-move and splicing forgeries.

Table 3
Performance evaluation of tampering detection

Type of Forgery	PoDA (%)	Accuracy	Precision	Recall	F1	Memory	PSNR	MSE
Copy Move	1.01	97.45	96.22	91.75	93.93	59.49	37.88	0.00915
	20.70	96.32	83.27	93.19	87.95	40.46	38.29	0.00171
	4.30	97.28	92.76	92.67	92.71	61.40	36.52	0.00814
	1.19	96.69	89.16	88.89	89.02	35.40	39.41	0.00346
	9.60	97.95	90.30	97.85	93.92	21.61	27.67	0.00284
Average	7.36	97.138	90.342	92.87	91.506	43.672	35.954	0.00506

Splicing	0.44	95.43	78.69	85.57	81.98	56.36	35.14	0.00664
	3.60	97.86	97.12	90.75	93.83	33.52	43.44	0.00250
	0.98	95.38	99.97	95.15	91.96	39.90	34.34	0.01057
	35.22	96.89	86.76	89.97	88.34	28.18	26.46	0.00940
	51.95	97.11	85.45	91.19	99.23	54.52	33.20	0.01090
Average	18.438	96.534	89.598	90.526	91.068	42.496	34.516	0.008

The results show that both Copy-Move and Splicing forgeries are detected with high accuracy, which are 95% to 98%. Copy-Move generally achieves higher precision and recall balance, while Splicing shows stronger performance in some cases but with more variation. PSNR values are moderate, which are 27 to 43 dB, and MSE stays low, indicating good image quality retention. Overall, the method is effective, with Copy-Move slightly more stable than Splicing.

4. Conclusion

This study successfully implemented the hybrid framework for image tampering detection and recovery by integrating Gauss Jordan Elimination (GJE), Singular Value Decomposition (SVD), and Local Binary Pattern (LBP) histogram. The framework successfully detected copy-move and splicing forgeries with high accuracy on the CASIA 2.0 dataset, while providing reliable recovery of tampered regions. The methods complemented each other, where GJE enabled pixel-level recovery, SVD captured structural inconsistencies, and LBP identified texture variations. The findings indicated the effectiveness of the proposed method compared to existing approaches, particularly in balancing detection and recovery. Overall, the framework provides a practical and efficient solution for digital image forensics, with potential to be extended into larger datasets and real-time applications in future work.

References

[1] Chakraborty, Sunen, Kingshuk Chatterjee, and Paramita Dey. "Detection of image tampering using deep learning, error levels and noise residuals." *Neural Processing Letters* 56, no. 2 (2024): 112. https://doi.org/10.1007/s11063-024-11448-9

[2] Chen, Haipeng, Xiwen Yang, and Yingda Lyu. "Copy-move forgery detection based on keypoint clustering and similar neighborhood search algorithm." *IEEE Access* 8 (2020): 36863-36875. https://doi.org/10.1109/ACCESS.2020.2974804

[3] Chennupati, Om Sai Teja. "A structured approach to JPEG tampering detection using enhanced fusion algorithm." PhD diss., Tesi di laurea mag. Computer Science: Blekinge Institute of Technology, 2021.

[4] Dixit, Anuja, and Soumen Bag. "A fast technique to detect copy-move image forgery with reflection and non-affine transformation attacks." *Expert Systems with Applications* 182 (2021): 115282. https://doi.org/10.1016/j.eswa.2021.115282

[5] Fernández, Edgar González, Ana Lucila Sandoval Orozco, and Luis Javier García Villalba. "A multi-channel approach for detecting tampering in colour filter images." *Expert Systems with Applications* 230 (2023): 120498. https://doi.org/10.1016/j.eswa.2023.120498

[6] Ganokratanaa, Thittaporn, Manotham Damnnoen, Puchit Katesomboon, Phacharaphon Aiamphan, Korrapant Maneeta, and Warin Wattanapornprom. "Human vs. AI: Leveraging Machine Learning and Deep Learning to Verify Image Authenticity." In *2024 28th International Computer Science and Engineering Conference (ICSEC)*, pp. 1-6. IEEE, 2024. https://doi.org/10.1109/ICSEC62781.2024.10770741

[7] Babu, SBG Tilak, and Ch Srinivasa Rao. "An optimized technique for copy–move forgery localization using statistical features." *ICT Express* 8, no. 2 (2022): 244-249. https://doi.org/10.1016/j.icte.2021.08.016

[8] Kumar, Sanjeev, Suneet K. Gupta, Manjit Kaur, and Umesh Gupta. "VI-NET: a hybrid deep convolutional neural network using VGG and inception V3 model for copy-move forgery classification." *Journal of Visual Communication and Image Representation* 89 (2022): 103644. https://doi.org/10.1016/j.jvcir.2022.103644

[9] Luo, Yuling, Liangjia Li, Junxiu Liu, Shunbin Tang, Lvchen Cao, Shunsheng Zhang, Senhui Qiu, and Yi Cao. "A multi-scale image watermarking based on integer wavelet transform and singular value decomposition." *Expert Systems with Applications* 168 (2021): 114272. https://doi.org/10.1016/j.eswa.2020.114272

[10] Molina-Garcia, Javier, Beatriz P. Garcia-Salgado, Volodymyr Ponomaryov, Rogelio Reyes-Reyes, Sergiy Sadovnychiy, and Clara Cruz-Ramos. "An effective fragile watermarking scheme for color image tampering detection and self-recovery." *Signal Processing: Image Communication* 81 (2020): 115725. https://doi.org/10.1016/j.image.2019.115725

[11] Nazir, Tahira, Marriam Nawaz, Momina Masood, and Ali Javed. "Copy move forgery detection and segmentation using improved mask region-based convolution network (RCNN)." *Applied Soft Computing* 131 (2022): 109778. https://doi.org/10.1016/j.asoc.2022.109778

[12] Yuan, Xiaochen, Xinhang Li, and Tong Liu. "Gauss–Jordan elimination-based image tampering detection and self-recovery." *Signal Processing: Image Communication* 90 (2021): 116038. https://doi.org/10.1016/j.image.2020.116038

[13] CASIA 2.0 Image Tampering Detection Dataset. (2021, March 22). Kaggle. https://www.kaggle.com/datasets/divg07/casia-20-image-tampering-detection-dataset

[14] Wang, Xiangyang, Wencong Chen, Panpan Niu, and Hongying Yang. "Image copy-move forgery detection based on dynamic threshold with dense points." *Journal of Visual Communication and Image Representation* 89 (2022): 103658. https://doi.org/10.1016/j.jvcir.2022.103658

[15] Zhong, Jun-Liu, and Chi-Man Pun. "Two-pass hashing feature representation and searching method for copy-move forgery detection." *Information Sciences* 512 (2020): 675-692. https://doi.org/10.1016/j.ins.2019.09.085