Journal of Applied Science, Engineering, Technology, and Education *ISSN: 2685-0591 (online)*
Vol. 7 No. 2 (2025)
https://doi.org/10.35877/454RI.asci3939

Modified SIFT-Based Kirsch Edge Detection Approach for Copy-Move Forgery Detection

Bashir Idris[a,b,*], Lili N. Abdullah[b], Alfian Abdul Halin[b], & Mohd Taufik Abdullah Selimun[b]

[a]*Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia*
[b]*School of Secondary Education (Sciences), Federal College of Education (Technical) Gusau, PMB 1088 Zamfara State, Nigeria*

Abstract

Copy-move forgery (CMF) is one of the most common and challenging image forgeries due to its seamless duplication. This paper proposes a passive detection method that combines a modified Kirsch (mKirsch) edge detector with a novel SIFT-based descriptor (DivSIFT) to effectively identify and localize CMF. The mKirsch detector enhances edge sensitivity by removing specific directional masks, boosting the quality of keypoints extracted by DivSIFT. Experiments were conducted on three benchmark datasets; MICC-F220, CoMoFoD, and MICC-F8Multi under various attack conditions including rotation, scaling, JPEG compression, and multiple cloning. The proposed method achieved high accuracy, notably reaching a 90.91% true positive rate (TPR), 100% precision, and a 95.24% F-measure when NE_SE or SW_NW masks were removed. It also maintained robustness under rotation (81.82% TPR) and scaling (96.97% TPR). Compared to state-of-the-art methods, our approach achieved a lower false positive rate (0%) and faster execution time (2.74 seconds), demonstrating its practical value in real-world forensics.

Keywords: Copy-move forgery detection, image forensics, digital image forgery, Kirsch edge detector, SIFT descriptor.

Received: 20 March 2025 Revised: 30 May 2025 Published: 31 August 2025

1. Introduction

The widespread availability of affordable, high-resolution cameras and advanced, user-friendly devices like smartphones has led to an increase in the use of digital images in recent years. These images offer significant advantages in areas such as modern media (both print and digital), security surveillance, insurance claims, and medical diagnoses. However, despite these benefits, digital images are increasingly being manipulated to spread false information, which raises concerns about their authenticity, especially from a forensic standpoint. People at different levels are exploiting images for malicious purposes. For example, Figures 1 and 2 depict images used in advertising strategies and political campaigns. Figure 1 shows a controversial election photo of John Kerry, where he was digitally altered to appear seated next to Jane Fonda at an anti-war rally in 1970, giving a misleading impression of his political stance. In Figure 2, a marketing campaign involved swapping the head of a Black man with that of a white man (Zheng et al., 2019).

The study of image forgery became more serious around 2001 (Fridrich et al., 2003). Two of the most common and troubling types of image manipulation, which are difficult for the human eye to detect, are copy-move forgery and splicing (Asghar et al., 2016; Rao & Ni, 2017). This research focuses specifically on detecting copy-move forgery. A number of techniques were proposed for detecting such forgery in digital images (Dixit & Naskar, 2017; Soni et al., 2019), including both active and passive methods (Joglekar & Chatur, 2015). The active technique assumes that the image contains a signature or watermark, and the presence of forgery is determined by comparing the extracted features with the original ones. However, this approach has significant limitations because many images lack any prior signatures or watermarks (Korus, 2017), restricting the method's applicability. As a result, passive detection solutions have been developed, which work effectively irrespective of any prior information about the image. These methods extract key features from the image and use statistical analysis to determine if the image has been altered

* Corresponding author.
E-mail address: xxxx@xxxxxx.edu

(Chien-chang Chen et al., 2019). Additionally, images affected by copy-move forgery often involve pixel modifications, typically through geometric transformations and post-processing (Chenglong Chen et al., 2017). This paper proposes a passive detection approach for identifying and locating forged regions in digital images, particularly those with geometrically altered pixels, using modified SIFT descriptors and the Kirsch edge detector.

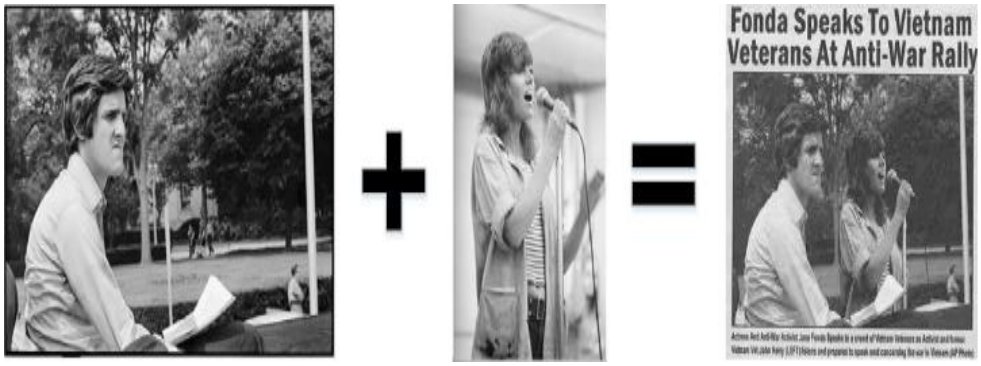

Fig. 1. A software company used a marketing tactic where a Black man's head (left) was replaced with that of a white man (right).

Fig. 2. The Kerry-Fonda photo controversy involved a manipulated image that falsely depicted John Kerry alongside Jane Fonda at a 1970s anti-war rally, creating a misleading political narrative.

2. Related work

With the increasing prevalence of image manipulation, researchers have explored various methods to detect digital forgeries. Among these, block-based techniques involve segmenting an image into overlapping or non-overlapping blocks and extracting relevant features from each segment. These features are then compared using different matching strategies. Initially, approaches such as exhaustive search, exact match, auto-correlation, and robust match were employed for feature comparison. While these methods successfully identified forgeries, they also produced a high number of false matches. To improve accuracy, alternative techniques such as lexicographic sorting, radix sorting, hash values, k-d trees, and Euclidean distance-based matching have been introduced (Teerakanok & Uehara, 2019).

Early research on CMF detection utilized block-based methods, with Fridrich et al. being the first to apply this approach. Their technique involved dividing an image into fixed-size overlapping blocks, analyzing them using the Discrete Cosine Transform (DCT), and subsequently sorting them lexicographically for feature matching (Fridrich et al., 2003). Many later studies built upon this foundation, aiming to enhance efficiency and reduce computational complexity. Researchers have explored different image representation domains, such as intensity-based (Wang et al., 2009), frequency-based (Bayram, 2009), and moment-based (Baumeister et al., 2005) techniques. Cozzolino et al. introduced a more efficient method leveraging the PatchMatch algorithm, which significantly improved detection performance. However, challenges such as high computational costs and limited robustness to affine transformations persisted (Cozzolino et al., 2014). More recently, Pavlović et al. (2019) proposed a non-overlapping block-based approach that extracted multifractal features and applied metaheuristic classification. Although the method successfully detected forgeries, it struggled with images subjected to geometric transformations. Hosny et al. developed the QPCEMT-based method, designed to work with sub-sampled images. While effective for such images, its performance declined when dealing with dense and smooth-textured regions (Hosny et al., 2019). Another approach by Gani & Qadir (2020) involved segmenting an image into blocks, applying DCT for feature extraction, and using cellular automata for duplication detection. Although this method accurately identified forgeries, its

computational complexity remained a significant drawback. Overall, block-based techniques continue to face challenges related to processing complexity, geometric transformations, and resistance to various forgery techniques.

Keypoint-based techniques address some of the limitations of block-based methods by focusing on high-entropy key points within an image rather than segmenting them into blocks. These methods extract distinctive features from key points and use them for forgery detection, avoiding the need for exhaustive block matching (Hegazi et al., 2021; Prakash et al., 2019). One of the most widely used keypoint-based approaches is the Scale Invariant Feature Transform (SIFT), known for its effectiveness in detecting copy-move forgeries (Warif et al., 2016). Amerini et al. (2011) utilized SIFT to identify multiple copy-move forgeries, employing the g2NN matcher to determine feature similarities. To reduce false positives, they later incorporated Agglomerative Hierarchical Clustering (AHC) and introduced the J-Linkage clustering algorithm to refine matching accuracy (Amerini et al., 2013). Other researchers, such as Pan & Lyu (2010), combined the SIFT descriptor with correlation maps to detect forgeries. Since SIFT is scale-invariant, it effectively identifies forgeries even in the presence of geometric distortions and post-processing effects.

However, its performance degrades when handling blurs, flipping, and smoothing operations. Zhu et al. (2015) applied ORB keypoint detection and RANSAC filtering to reduce false matches, making their method effective against geometric transformations. Nevertheless, high-resolution images posed computational challenges. Zhou et al. (2017) introduced a color-invariant model and the SURF detector for feature extraction, further refining the key point selection process for improved forgery detection. Chou & Lee (2018) developed a multi-scale feature extraction technique, applying SIFT over non-overlapping irregular patches and matching features to identify duplicates. Similarly, Alberry et al. (2018) combined the SIFT algorithm with Fuzzy C-Means clustering, enhancing the detection of geometrically transformed images but struggling with post-processed ones. Yang et al. (2018) proposed a keypoint distribution strategy to ensure even extraction from smooth regions before applying SIFT, which proved effective for rotated images but not for affine transformations. Liu et al. (2019) combined Local Intensity Order Patterns (LIOP) with SIFT for keypoint extraction, using a 2gNN matcher alongside density grid-based filtering to minimize false positives. Niyishaka & Bhagvati (2020) adopted BRISK keypoint detection based on Sobel edges and clustered results using Blobs. While this method effectively distinguished original and duplicated regions, the sensitivity of the Sobel edge detector to noise sometimes led to incorrect edge detection. Jiang et al., (2024) proposed a strategy to extract a large number of keypoints, minimizing the risk of missed detections. To efficiently handle the increased keypoint set, a group matching algorithm is applied, streamlining the matching process and maintaining detection accuracy.

Traditional computer vision methods for keypoint detection often depend on histogram-based feature descriptors, with SIFT being one of the most well-known (Lowe, 2004). Alternative techniques employ various feature extraction methods, including kernel convolution (Mukundan et al., 2017), comparing intensity values (Leutenegger et al., 2011), using Haar wavelets (Bay et al., 2008), and analyzing pixel arrangements (Wang et al., 2016).

To enhance SIFT's robustness and efficiency, researchers have developed various modifications. Bellavia & Colombo (2020) reviewed these advancements, including the RootSIFT algorithm, which replaces Euclidean distance with the Hellinger Kernel for improved histogram-based similarity measurement. Arandjelovic & Zisserman (2012) introduced RootSIFT by normalizing SIFT vectors using L1 normalization followed by a square-root transformation, demonstrating its effectiveness as a refined version of SIFT. RootSIFT has since been widely adopted in object detection and forgery detection tasks. Further enhancements, such as RootSIFT-PCA, have also been proposed, incorporating Principal Component Analysis (PCA) for feature compression (Bursuc et al., 2015).

3. Proposed Methodology

DivSIFT and mKirsch are the proposed methods. DivSFT modification is routed from the idea in RooSIFT (Arandjelovic & Zisserman, 2012). The mKirsch on the other hand is modified by reducing the eight-edge filters to six, in order to secure distinctive and strong image edges. These procedures were accomplished in four steps as categorized in Fig. 3.

3.1. Pre-processing

The method presented is a passive image forgery detection technique that processes input in the form of RGB images. These images are initially transformed into grayscale using the formula in (Eq.1), which simplifies the processing by reducing the color channels. Once converted, the grayscale image is utilized to extract key points through SIFT and detect blobs using the mKirsch edge method.

$$\text{Gray} = 0.299 \times R + 0.587 \times G + 0.114 \times B \quad (1)$$

3.2. DivSIFT Descriptor

In the RootSIFT approach, the SIFT vector is first L1-normalized, then each element is square-rooted using the Hellinger kernel (Arandjelovic & Zisserman, 2012), which enhances performance compared to the standard Euclidean-based SIFT (Lowe, 2004). Despite this, the square root operation tends to be computationally intensive due to factors like function call overhead. While division may take longer during execution, it is generally less demanding on system resources. To address this, the proposed method replaces the square root operation with division, aiming for a balance between accuracy and computational efficiency (Bursuc et al., 2015). This alternative approach consists of two key steps:

- Applying L1 normalization to the original L2-normalized SIFT vector
- Performing an element-wise division of the L2-normalized vector by the L1-normalized vector

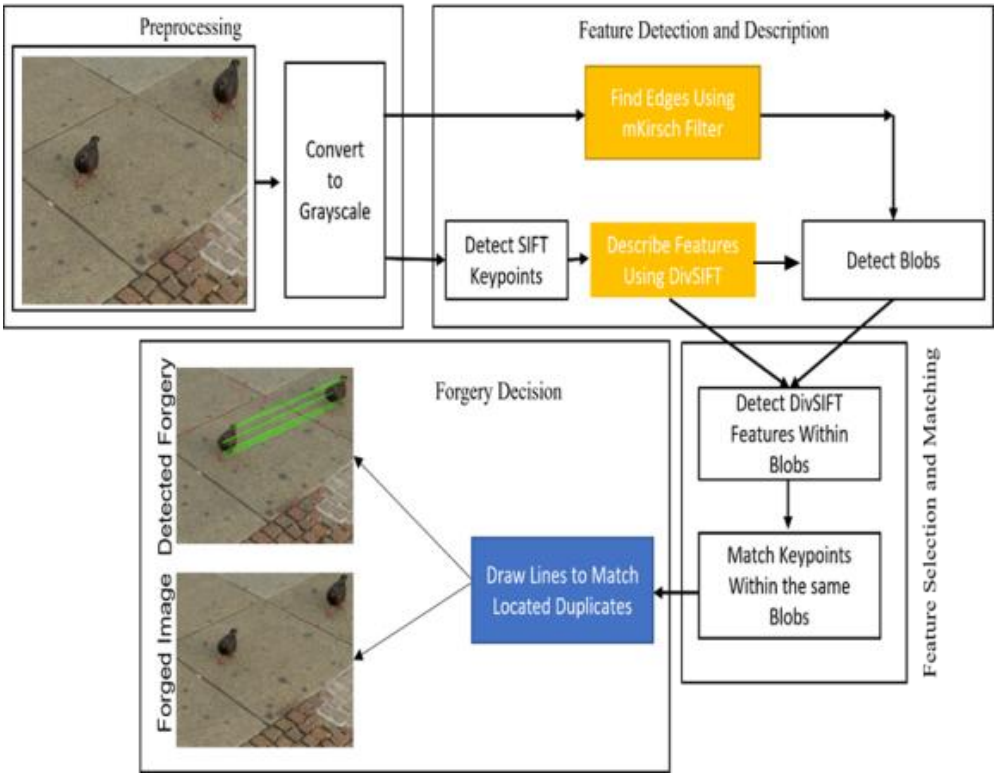

Fig. 3. Workflow of the Proposed Method

The resulting vector, called DivSIFT, is created by averaging the elements of the original vector after it has been L2-normalized. DivSIFT modifies the Hellinger Kernel method (Arandjelovic & Zisserman, 2012) by using division instead of the square root in the calculation, as shown below:

The Hellinger Kernel**,**

$$H(x, y) = \sum_{i=1}^{n} \sqrt{x_i y_i} \quad (2)$$

Replacing the Euclidian distance term in the Helinger kernel, in (Eq. 2), resulted in (Eq. 3)

$$H(x, y)^2 = \|x\|_2^2 + \|y\|_2^2 - 2SE(x, y) \quad (3)$$

Given $\|x\|_2^2 = 1 \; and \; \|y\|_2^2 = 1$, it implies to (eq. 4 and 5).

$$H(x, y)^2 = 1 + 1 - 2SE(x, y) \quad (4)$$

Simplify:

$$H(x,y)^2 = 2 - 2SE(x,y) \tag{5}$$

By replacing the square root with the division operation, (Eq.1) becomes (Eq. 6).

$$H(x,y) = \sum_{i=1}^{n} \frac{x_i}{\sqrt{y_i}} \tag{6}$$

Previously, replacing the Euclidean distance with the Hellinger function led to (Eq. 3). In contrast, using a division operation instead produces (Eq. 7).

$$d_H(x,y)^2 = \|x\|_2^2 + \|y\|_2^2 - 2\sum_{i=1}^{n} \frac{x_i}{\sqrt{y_i}} \tag{7}$$

Since $\|x\|_2^2 = 1 \; and \; \|y\|_2^2 = 1$ then, the equation becomes (Eq. 8 and 9) as the transformed kernel.

$$d_H(x,y)^2 = 2 - 2\sum_{i=1}^{n} \frac{\sqrt{x_i}}{\sqrt{y_i}}; \tag{8}$$

$$d_H(x,y) = \sum_{i=1}^{n} \frac{x_i}{\sqrt{y_i}}. \tag{9}$$

3.3. Advantage of Division Replacement

The rationale for using division instead of the square root in the Hellinger kernel can be understood in terms of the mathematical and statistical properties introduced by the transformation (Hastie et al., 2009; Genton, 2001). However, looking at it in the context of Relative differences in magnitude, linearity in the numerator, the impact of zero components, and flexibility with normalization can provide a comprehensive overview of the rationale behind using the division.

Emphasis on Differences in Magnitude. In the Original Kernel (Square Root), (eq. 2), the square root operation tends to smooth out differences between the components. x_i and y_i, since the square root is a concave function that grows more slowly than the input values. This can reduce the impact of large differences between x_i and y_i. However, the transformed kernel, in (eq. 9), proposed a division operation that emphasized differences, especially when x_i is small relative to y_i. This makes the kernel more sensitive to variations in the components of the vectors, highlighting relative differences more strongly (Pardo, 2019; Shawe-Taylor & Cristianini, 2004). In the case of linearity in the Numerator, the product $\sqrt{x_i y_i}$ introduces non-linearity in both x_i and y_i. While the transformed kernel $\frac{x_i}{\sqrt{y_i}}$ retains linearity in x_i Simplifying certain types of analysis and making the contribution of each component of *x* directly proportional to its value, adjusted by *y* (Schölkopf & Smola, 2002). Also on the impact of Zero Components, the original kernel is sensitive to zero components in that, if $y_i = 0$, the product $\sqrt{x_i y_i}$ becomes zero, making the contribution of that term to the sum zero. While with the transformed kernel, if $y_i = 0$, the term $\frac{x_i}{\sqrt{y_i}}$ becomes undefined. This implicitly ignores components of *x* where *y* is zero. This can be useful in cases where zero components in *y* are intended to mask or ignore the corresponding components in x (Genton, 2001). Flexibility with Normalization. In the original kernel, the Hellinger kernel is closely tied to normalized vectors, typically used with probability distributions where the components sum to 1. While the Transformed Kernel's division operation doesn't inherently assume normalized vectors, providing more flexibility in how the kernel can be applied. This can be advantageous in contexts where normalization is either not possible or not desired (Hastie et al., 2009). The original kernel is suitable for comparing probability distributions and other normalized data. Captures non-linear relationships well, making it useful in statistical and probabilistic contexts. The transformed kernel on the other hand is useful in scenarios where relative differences between feature magnitudes are important. And can be applied in machine learning tasks, such as classification or regression, where one set of features (represented by *y* serves as a reference or baseline (Bishop, 2006; Shawe-Taylor & Cristianini, 2004).

In summary, the primary advantage of the transformed kernel is its emphasis on relative differences and its sensitivity to the magnitude of components, which can provide a different perspective on the relationship between the vectors *x* and *y*. This transformation changes the nature of the similarity measure, making it potentially more suitable for applications where the relative sizes of vector components carry significant meaning. As a result, the developed CMFD detection method utilizes the newly introduced DivSIFT descriptor to identify unique image features. These features are then employed in the CMFD process using the subsequent stages.

3.4. DivSIFT Feature Generations

The DivSIFT descriptor is employed to extract distinctive features from local regions of an image, using the foundation of the SIFT algorithm. SIFT is known for its robustness against changes in scale, rotation, lighting conditions, noise, distortion, and varying viewpoints. According to Lowe (2004), the SIFT process involves five key stages: detection of extrema in scale space, keypoint localization, orientation assignment, descriptor construction, and keypoint matching.

In the proposed method, during the descriptor computation phase, the standard SIFT descriptor from Lowe's method is replaced with the DivSIFT descriptor. This results in the generation of a 128-dimensional feature vector, which is used for matching during the forgery detection process. Fig. 4 illustrates the key points and descriptors obtained using the DivSIFT approach, while Algorithm 1 outlines the steps involved in implementing the DivSIFT algorithm.

3.5. Proposed mKirsch Edge Detector

The modified Kirsch edge detection method (mKirsch) enhances both feature detection and the accuracy of key point matching. Traditionally, the Kirsch edge detector employs eight directional masks to identify the strongest edge responses. In this study, the method is refined by reducing the number of masks from eight to six. This adjustment retains the vertical and horizontal edge detection masks in both directions while selectively removing diagonal masks either one from each diagonal direction or two from a single diagonal orientation. For instance, masks corresponding to directions like north-east and south-east, or combinations such as north-west and south-west, may be excluded, as illustrated in Fig. 5.

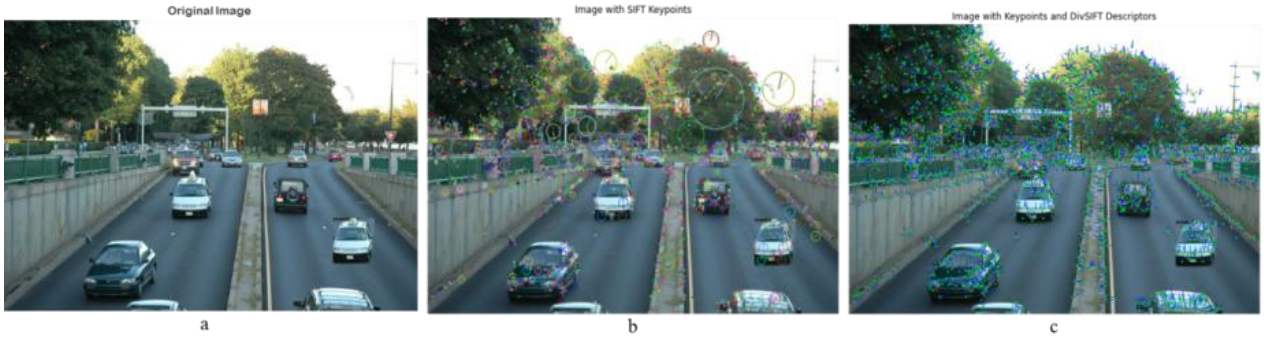

Fig. 4. DivSIFT Feature Generations; a) The original image, b) Points identified using SIFT, c) Descriptors generated using DivSIFT.

Algorithm 1: Proposed DivSIFT implementation algorithm

Input: An image I(x,y)
Output: Keypoints (kps) and descriptors (descs)
for each of the **N** images in the dataset:
    Convert the image I(x,y) to grayscale, resulting in GI(x,y)
    Detect key points and compute descriptors
    If no key points are found, return an empty set
    Otherwise:
        Apply L1 normalization to the descriptors to obtain a modified Hellinger representation.
        for each descriptor in the set:
            Compute the L1 norm along axis 1 while retaining dimensions
            Normalize the descriptor by dividing it by its corresponding L1 norm
        return the key points and the newly generated **DivSIFT** descriptors

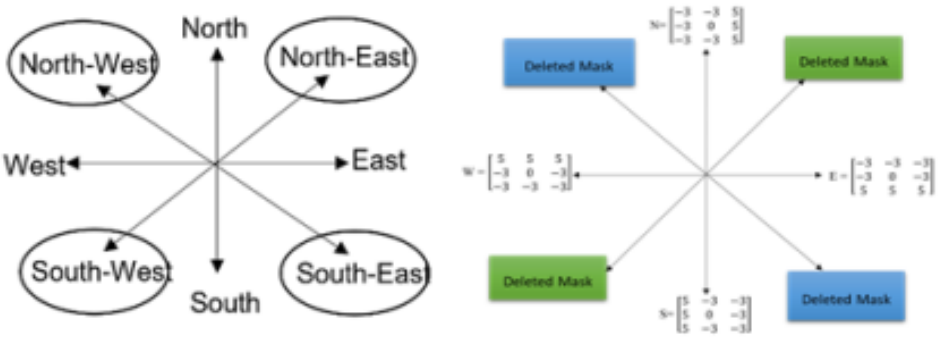

Fig. 5. Kirsch Filter Directions and Diagonal filters to be Deleted (Venmathi et al., 2016)

The proposed mKirsch edge detector focuses on enhancing edge detection by targeting vertical (North-South) and horizontal (East-West) orientations, while selectively applying diagonal masks such as North-East with South-West or North-West with South-East. As illustrated in Fig. 5(b), specific diagonal masks were deliberately omitted to optimize the original Kirsch filter improving processing speed at the cost of a reduced number of detected edges. However, this trade-off results in better segmentation quality and more effective feature extraction. Figure 6 and Table 1 present the performance evaluation of four mKirsch variations, each removing different combinations of directional masks. The first group removed diagonals in opposite directions (e.g., North-East and South-East or North-West and South-West), while the second excluded masks along a single diagonal axis (e.g., North-East and South-West or North-West and South-East). Across all tests, configurations with selective mask deletion consistently outperformed the unmodified version, indicating that removing less informative edge directions reduces noise and enhances detection accuracy within the CMFD pipeline.

Visual comparisons (Fig. 6) confirm these findings. In particular, **SW_NW_Deleted** and **NE_SE_Deleted** configurations produced cleaner and more distinct edge maps by suppressing noisy gradients often caused by background textures or compression artifacts. These configurations align better with tampering patterns, improving sensitivity to altered regions while minimizing false detections. Both configurations achieved top performance metrics, 90.91% TPR, 0% FPR, 100% Precision, 90.91% Recall, and a 95.24% F-Measure demonstrating a strong balance between accuracy and reliability. The refined edge maps generated enhance the CMFD process by preserving only the most relevant structural details. The selective removal of specific diagonal masks in mKirsch improves segmentation quality and detection precision, making **SW_NW_Deleted** and **NE_SE_Deleted** configurations highly effective for practical forgery detection applications. Algorithm 1 depicts the mKirsch implementation.

Fig. 6. Enhanced Kirsch Filter for Edge Detection: Emphasized Diagonal Operators

Algorithm: mKirsch Edge Detection Algorithm's Implementation

Input: RGB image I(x, y)
Output: Edge-enhanced grayscale image (mKirsch_Image)

```
    Convert to Grayscale:
        Gi ← Grayscale(I)
    Initialize Z_image:
    Z_image ← zeros_like(Gi)
    Select Diagonal Masks:
            if use_diagonals = "NW_SW":
                masks ← {North, South, East, West, North-West, South-West}
            elif use_diagonals = "NE_SE":
                masks ← {North, South, East, West, North-East, South-East}
            else:
                masks ← {North, South, East, West}  # No diagonals

    Apply Each Mask:
    for each mask M in masks:
            E ← Gi * M (convolution)
            for each pixel (i, j):
                    If E[i, j] is the max at (i, j) across all masks:
                        Z_image[i, j] += E[i, j]
    Normalize Output:
    For each pixel (i, j) in Z_image:
            If Z_image[i, j] > 255, set Z_image[i, j] = 255
    Return Output:
    mKirsch_Image ← Z_image
```

3.6. Blob Detection

After edge detection is performed using the mKirsch method by identifying the strongest response from its masks blobs are identified from the extrema of the filter responses across multiple scales. This is achieved by convolving the image with a blob filter at different scales. The final blob positions are approximated using the Difference of Gaussian (DoG) within the scale-space of the Laplacian of Gaussian. Points found within these blobs are then matched to identify duplicated regions, indicating copy-move forgery (CMF). The result of this process is illustrated in Fig 7.

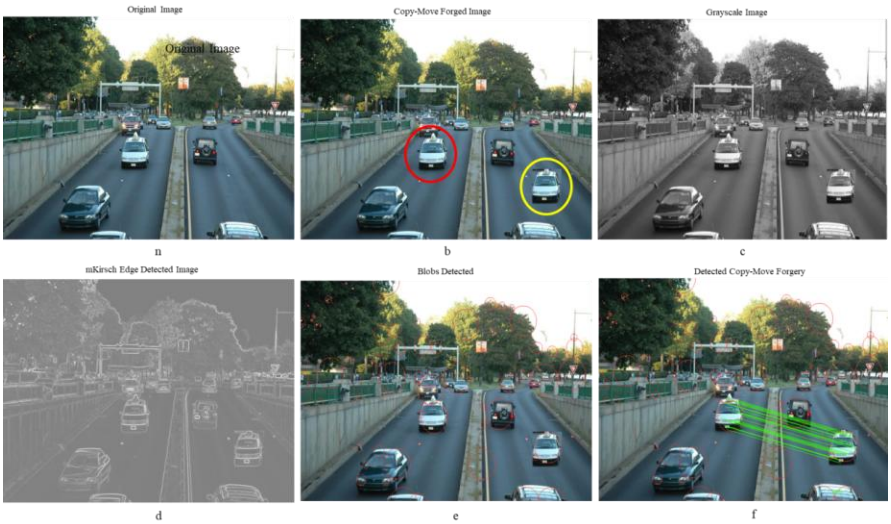

Fig. 7. Forgery Detection Using mKirsch and DivSIFT**:** (a) The original image, (b) The manipulated version, (c) Grayscale Conversion, (d) Edge detection using the mKirsch operator, (e) identification of blob regions, and (f) The final result highlighting the detected forgeries.

4. Results and Evaluation

This section evaluates the results obtained using the proposed techniques, focusing on their effectiveness as compared to current state-of-the-art methods. The analysis aims to assess the robustness of the new descriptor and the modified Kirsch filter.

4.1 Mask Deletion Impact on Edge Detection

To evaluate the effect of directional edge emphasis, we tested different configurations of the Kirsch and mKirsch detectors by deleting specific diagonal masks. This was intended to reduce noise and irrelevant edge responses, thereby enhancing CMFD precision. Table 1 presents the performance metrics for mask deletion experiments. The mKirsch detector achieved the highest performance when either the SW_NW or NE_SE directional masks were removed. This configuration resulted in TPR: 90.91%**,** FPR: 0%**,** Precision: 100%**,** and F-measure: 95.24%.

These findings suggest that modifying the edge response can significantly improve the ability to distinguish between authentic and duplicated regions.

Table 1. Performance Evaluation of the mKirsch Edge Detector with Various Mask Configurations

Method Variant	TPR (%)	FPR (%)	Precision (%)	Recall (%)	F-Measure (%)
NE_SE Masks Removed	**90.90**	**0**	**100**	**90.90**	**95.23**
NW_SE Masks Removed	81.81	0	100	81.81	90.00
NE_SW Masks Removed	86.36	0	100	86.36	92.68
SW_NW Masks Removed	**90.90**	**0**	**100**	**90.90**	**95.23**
Standard Kirsch (No Masks Removed)	81.81	0	100	81.81	90.00

4.2 CMFD Performance on Forgery Variants

We further evaluated six detection configurations: SIFT + Kirsch + Blob, SIFT + mKirsch + Blob, RootSIFT + Kirsch + Blob, RootSIFT + mKirsch + Blob, DivSIFT + Kirsch + Blob, and DivSIFT + mKirsch + Blob. These methods were tested across six major forgery scenarios as in Table 2.

Table 2. Performance Evaluation of the SIFT-Based Kirsch Edge Detection Approaches for Copy-Move Forgery Detection

Proposed Techniques	Plain CMF		Rotated CMF		Scaled CMF		Combined CMF		JPEG-Compressed CMF		Multiple CMF	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
Kirsch + SIFT + Blob	81.82	0.00	81.82	0.00	84.85	0.00	80.00	0.00	91.3	14.29	53.85	0.00
mKirsch + SIFT + Blob	100.00	0.00	81.82	0.00	96.97	0.00	90.91	0.00	94.03	14.29	61.54	0.00
Kirsch + RootSIFT + Blob	81.82	0.00	77.27	0.00	84.85	0.00	81.82	0.00	92.54	14.29	53.85	0.00
mKirsch + RootSIFT + Blob	100.00	0.00	77.27	0.00	93.94	0.00	86.36	0.00	92.54	12.9	53.85	0.00
Kirsch + DivSIFT + Blob	90.91	0.00	84.09	0.00	84.85	0.00	86.36	0.00	94.03	14.29	53.85	0.00
mKirsch + DivSIFT + Blob	100.00	0.00	81.82	0.00	96.97	0.00	90.91	0.00	95.52	12.31	61.54	0.00

4.3 Plain Copy-Move Forgery (CMF)

All configurations achieved 100% TPR, effectively detecting duplicated regions with no geometric transformation and post-processing. Zero false positives were recorded in each method. The result is visualized in Fig 8.

Fig. 8. Visual Illustration of Detection Accuracy on Plain Images

4.4 Rotation Variants (10°–40°)

The performance dropped slightly under rotation, but the DivSIFT + Kirsch + Blob method demonstrated the best robustness, achieving a TPR of 84.09% and maintaining 0% FPR**.** Fig. 9 presents a visual result of the detection performance.

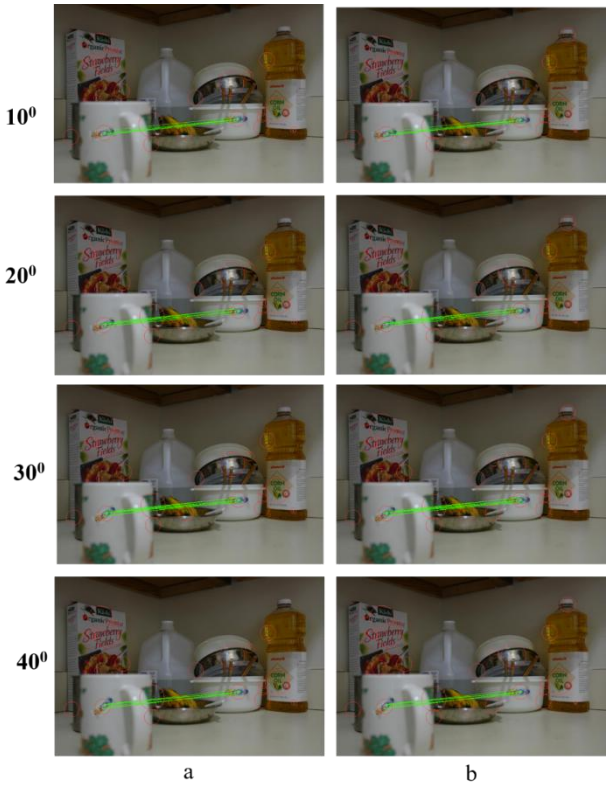

Fig. 9. Visual Illustration of Detection Accuracy on Rotated images at 10°, 20°, 30°, and 40°.

4.5 Scaling Attacks

When cloned regions were scaled (e.g., 1.2x), mKirsch-based methods maintained high TPRs up to 96.97% and zero false positives. The robustness to scaling demonstrates strong invariance features in DivSIFT and mKirsch edge detection. Fig. 10 presents the detection accuracy of these methods.

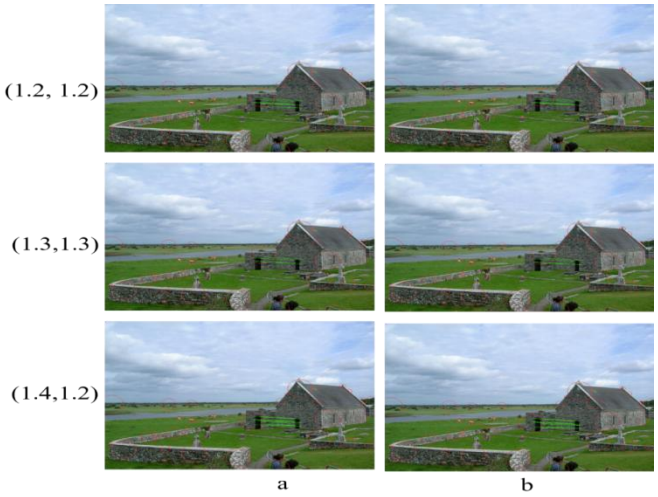

Fig. 10. Visual Illustration of Detection Accuracy on Scaled Images

4.6 Combined Rotation + Scaling (RS)

Performance under dual transformations remained high. The mKirsch + DivSIFT + Blob method consistently achieved zero FPR and TPRs above 90%**,** indicating strong resilience to geometric changes. Fig. 11 further illustrates the detection result of the RS attack.

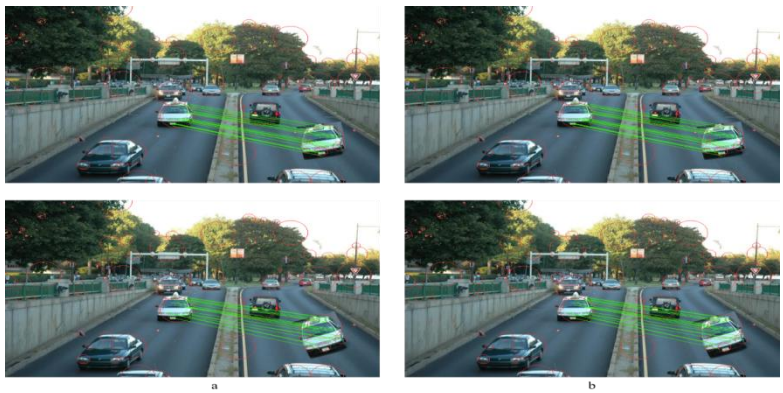

Fig. 11. Visual Illustration of Detection Accuracy on Scaled and Rotated Images

4.7 JPEG Compression

JPEG compression presents a common real-world challenge. Despite some degradation, mKirsch + DivSIFT + Blob maintained a TPR of 95.52%**.** However, FPR reached 12.31%**,** the highest among all tests, indicating a tradeoff between sensitivity and selectivity in lossy formats. The result is visualized in Fig. 12.

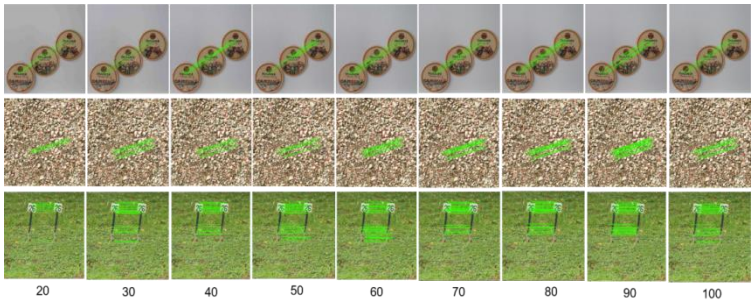

Fig. 12 . Visual Illustration of Detection Accuracy on JPEG Compressed Images at 20 to 100 quality factors

4.7 Multiple Clone Forgeries

In scenes with multiple duplicated regions, all methods registered 0% FPR**,** showing excellent precision. However, TPRs declined slightly (53.85–61.54%), reflecting the challenge in matching features across several instances with potential distortions. Fig. 13 shows some results of multiple clone forgery detection.

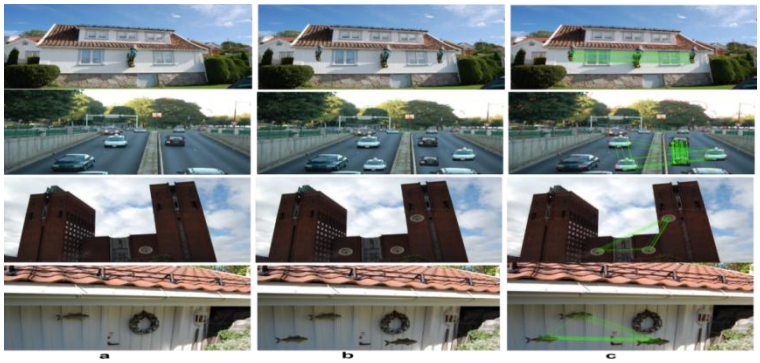

Fig. 13. Images Involving Copy-Move Forgery with Multiple Instances: a) Original Images, b) Forged Images with Copy-Move Manipulation, c) Detection Results Highlighting the Copy-Move Forgery

5. Discussion

5.1 Edge Detector and Descriptor Synergy

The proposed DivSIFT + Blob combination with mKirsch edge filtering delivered optimal performance across most conditions. The key contribution is the balance between TPR and FPR, maintaining zero false detections in most forgery scenarios while adapting to transformations. Blob filtering also contributed significantly by removing irrelevant features before keypoint detection and matching.

5.2 Challenges in Complex Cloning

While precision remained high in multiple clone forgeries, the TPR was moderately reduced. This could be attributed to increased complexity in spatial consistency checks and overlapping cloned features. Still, the zero FPR underscores its reliability for applications where false alarms are more critical than misses**,** such as academic integrity verification or legal forensics.

5.3 Comparison with State-of-the-Art CMFD Methods

Table 3 summarizes a comparative evaluation of our best method (DivSIFT + mKirsch + Blob) against state-of-the-art CMFD techniques from recent literature.

Table 3. Performance Evaluation: Proposed Method vs. State-of-the-Art Approaches

Method	True Positive Rate (TPR%)	False Positive Rate (FPR%)	F1-Score (%)	Execution Time (s)
Soni et al., 2019	89.75	12.54	–	1.37
Jaiswal et al., 2020	–	–	86.45	–
Tahaoglu et al., 2021	92.98	1.00	90.00	–
Yang et al., 2021	94.00	–	82.00	–
Niyishaka & Bhagvati, 2020	93.63	5.40	–	6.24
Alhaidery et al., 2023	93.75	7.25	93.75	31.14
Fu et al., 2023	–	–	95.12	54.23
Jiang et al., 2024	94.50	6.5		
Proposed Method	**90.00**	**0.00**	**94.73**	**2.74**

The proposed method offers a compelling mix of speed, precision, and reliability, making it practical for near real-time CMFD applications. Its **0**% FPR outperforms all other methods, many of which suffer from false detections due to high sensitivity.

6. Conclusion

This study proposed a CMFD technique combining DivSIFT descriptors, Blob filtering, and an enhanced mKirsch edge detector. Extensive experiments demonstrated (1) Excellent TPR under geometric transformations and compression. (2) Zero FPR in most forgery scenarios. And (3) Fast execution time (2.74 seconds), outperforming several recent methods.

While further improvement is needed in detecting multiple cloned regions, the approach provides a highly reliable and efficient solution suitable for image forensic authentication systems.

References

A. Alberry, H., A. Hegazy, A., & I. Salama, G. (2018). A fast SIFT-based method for copy move forgery detection. *Future Computing and Informatics Journal*, *3*(2), 159–165. https://doi.org/10.1016/j.fcij.2018.03.001

Alhaidery, M. M. A., Taherinia, A. H., & Shahadi, H. I. (2023). A robust detection and localization technique for copy-move forgery in digital images. *Journal of King Saud University-Computer and Information Sciences*, *35*(1), 449-461.

Amerini, I., Ballan, L., Caldelli, R., Del Bimbo, A., & Serra, G. (2011). A SIFT-based forensic method for copy-move attack detection and transformation recovery. *IEEE Transactions on Information Forensics and Security*, *6*(3 PART 2), 1099–1110. https://doi.org/10.1109/TIFS.2011.2129512

Arandjelovic, R., & Zisserman, A. (2012). Three things everyone should know to improve object retrieval. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2911–2918. https://doi.org/10.1109/CVPR.2012.6248018

Asghar, K., Habib, Z., & Hussain, M. (2016). Copy-move and splicing image forgery detection and localization techniques : a review. *Australian Journal of Forensic Sciences*, *0618*(May), 0–27. https://doi.org/10.1080/00450618.2016.1153711

Bayram. (2009). AN EFFICIENT AND ROBUST METHOD FOR DETECTING COPY-MOVE FORGERY Sevinc Bayram Polytechnic Institute of NYU ECE Dept . Brooklyn , NY Husrev Taha Sencar TOBB University of Economics & Technology Ankara , TURKEY Nasir Memon Polytechnic Insitute of NYU CIS Dep. *Image (Rochester, N.Y.)*, 1053–1056.

Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning* (Vol. 4, No. 4, p. 738). New York: springer.

Bellavia, F., & Colombo, C. (2020). RootsGloh2: Embedding RootSIFT “square rooting” in sGLOH2. *IET Computer Vision*, *14*(4), 138–143. https://doi.org/10.1049/iet-cvi.2019.0716

Bursuc, A., Tolias, G., & Jégou, H. (2015). Kernel local descriptors with implicit rotation matching. *ICMR 2015 - Proceedings of the 2015 ACM International Conference on Multimedia Retrieval*, 595–598. https://doi.org/10.1145/2671188.2749379

Chen, Chenglong, Ni, J., Shen, Z., & Shi, Y. Q. (2017). Blind Forensics of Successive Geometric Transformations in Digital Images Using Spectral Method: Theory and Applications. *IEEE Transactions on Image Processing*, *26*(6), 2811–2824. https://doi.org/10.1109/TIP.2017.2682963

Chen, Chien-chang, Lu, W., & Chou, C. (2019). *Rotational copy-move forgery detection using SIFT and region growing strategies*. *151*.

Chou, C. L., & Lee, J. C. (2018). Copy-move forgery detection based on local gabor wavelets patterns. *Advances in Intelligent Systems and Computing*, *733*, 47–56. https://doi.org/10.1007/978-3-319-76451-1_5

Cozzolino, D., Poggi, G., & Verdoliva, L. (2014). COPY-MOVE FORGERY DETECTION BASED ON PATCHMATCH Davide Cozzolino , Giovanni Poggi , Luisa Verdoliva Universit ´ a Federico II di Napoli , DIETI , 80125 Naples Italy. *International Conference on Image Processing(ICIP)*, 5312–5316.

Dixit, R., & Naskar, R. (2017). Review, analysis and parameterisation of techniques for copy-move forgery detection in digital images. *IET Image Processing*, *11*(9), 746–759. https://doi.org/10.1049/iet-ipr.2016.0322

Fu, G., Zhang, Y., & Wang, Y. (2023). Image copy-move forgery detection based on fused features and density clustering. *Applied Sciences*, *13*(13), 7528.

Fridrich, J., Soukal, D., & Lukáš, J. (2003). Detection of Copy-Move Forgery in Digital Images. *Proceedings of Digital Forensic Research Workshop, Cleveland, OH*, 1163–1168. https://doi.org/10.1109/ICMLA.2015.137

Gani, G., & Qadir, F. (2020). A robust copy-move forgery detection technique based on discrete cosine transform and cellular automata. *Journal of Information Security and Applications*, *54*, 102510. https://doi.org/10.1016/j.jisa.2020.102510

Genton, M. G. (2001). Classes of kernels for machine learning: a statistics perspective. *Journal of machine learning research*, *2*(Dec), 299-312.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer.

Hegazi, A., Taha, A., & Selim, M. M. (2021). An improved copy-move forgery detection based on density-based clustering and guaranteed outlier removal. *Journal of King Saud University - Computer and Information Sciences*, *33*(9), 1055–1063. https://doi.org/10.1016/j.jksuci.2019.07.007

Hosny, K. M., Hamza, H. M., & Lashin, N. A. (2019). Copy-for-duplication forgery detection in colour images using QPCETMs and sub-image approach. *IET Image Processing*, *13*(9), 1437–1446. https://doi.org/10.1049/iet-ipr.2018.5356

Jiang, L., Lu, Z., Gao, Y., & Wang, Y. (2024). Image Copy-Move Forgery Detection and Localization Scheme: How to Avoid Missed Detection and False Alarm. *arXiv preprint arXiv:2406.03271*.

Joglekar, N. P., & Chatur, P. N. (2015). A Compressive Survey on Active and Passive Methods for Image Forgery Detection. *International Journal Of Engineering And Computer Science*, *4*(1), 10187–10190.

Korus, P. (2017). Digital image integrity – a survey of protection and verification techniques. *Digital Signal Processing: A Review Journal*, *71*, 1–26. https://doi.org/10.1016/j.dsp.2017.08.009

Liu, K., Lu, W., Lin, C., Huang, X., Liu, X., Yeung, Y., & Xue, Y. (2019). Copy move forgery detection based on keypoint and patch match. *Multimedia Tools and Applications*, *78*(22), 31387–31413. https://doi.org/10.1007/s11042-019-07930-5

Lowe, D. G. (2004). *Distinctive Image Features from Scale-Invariant Keypoints*. *60*(2), 91–110.

Mikolajczyk, K. (2002). Detection of local features invariant to affine transformations Application to matching and recognition. *Most*, 171.

Niyishaka, P., & Bhagvati, C. (2020). Copy-move forgery detection using image blobs and BRISK feature. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-020-09225-6

Pan, X., & Lyu, S. (2010). Region duplication detection using image feature matching. *IEEE Transactions on Information Forensics and Security*, *5*(4), 857–867. https://doi.org/10.1109/TIFS.2010.2078506

Pardo, L. (2018). *Statistical inference based on divergence measures*. Chapman and Hall/CRC.

Pavlović, A., Glišović, N., Gavrovska, A., & Reljin, I. (2019). Copy-move forgery detection based on multifractals. *Multimedia Tools and Applications*, *78*(15), 20655–20678. https://doi.org/10.1007/s11042-019-7277-1

Prakash, C. S., Panzade, P. P., Om, H., & Maheshkar, S. (2019). Detection of copy-move forgery using AKAZE and SIFT keypoint extraction. *Multimedia Tools and Applications*, *78*(16), 23535–23558. https://doi.org/10.1007/s11042-019-7629-x

Rao, Y., & Ni, J. (2017). A deep learning approach to detection of splicing and copy-move forgeries in images. *8th IEEE International Workshop on Information Forensics and Security, WIFS 2016*, 1–6. https://doi.org/10.1109/WIFS.2016.7823911

Soni, B., Das, P. K., & Thounaojam, D. M. (2019). Geometric transformation invariant block based copy-move forgery detection using fast and efficient hybrid local features. *Journal of Information Security and*

Applications, *45*, 44–51. https://doi.org/10.1016/j.jisa.2019.01.007

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge university press.

Teerakanok, S., & Uehara, T. (2019). Copy-Move Forgery Detection: A State-of-the-Art Technical Review and Analysis. *IEEE Access*, *7*, 40550–40568. https://doi.org/10.1109/ACCESS.2019.2907316

Wang, J., Liu, G., Li, H., Dai, Y., & Wang, Z. (2009). Detection of image region duplication forgery using model with circle block. *1st International Conference on Multimedia Information Networking and Security, MINES 2009*, *1*, 25–29. https://doi.org/10.1109/MINES.2009.142

Wang, Z., Fan, B., Wang, G., & Wu, F. (2016). Exploring Local and Overall Ordinal Information for Robust Feature Description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(11), 2198–2211. https://doi.org/10.1109/TPAMI.2015.2513396

Warif, N. B. A., Wahab, A. W. A., Idris, M. Y. I., Ramli, R., Salleh, R., Shamshirband, S., & Choo, K. K. R. (2016). Copy-move forgery detection: Survey, challenges and future directions. *Journal of Network and Computer Applications*, *75*, 259–278. https://doi.org/10.1016/j.jnca.2016.09.008

Yang, B., Sun, X., & Guo, H. (2018). A copy-move forgery detection method based on CMFD-SIFT. *Multimedia Tools and Applications*, *1800*, 837–855. https://doi.org/10.1007/s11042-016-4289-y

Zheng, L., Zhang, Y., & Thing, V. L. L. (2019). A survey on image tampering and its detection in real-world photos. *Journal of Visual Communication and Image Representation*, *58*(December), 380–399. https://doi.org/10.1016/j.jvcir.2018.12.022

Zhou, Z., Wang, Y., Wu, Q. M. J., Yang, C. N., & Sun, X. (2017). Effective and Efficient Global Context Verification for Image Copy Detection. *IEEE Transactions on Information Forensics and Security*, *12*(1), 48–63. https://doi.org/10.1109/TIFS.2016.2601065

Zhu, Y., Shen, X., & Chen, H. (2015). *Copy-move forgery detection based on scaled ORB*. *2699*. https://doi.org/10.1007/s11042-014-2431-2