Forensic Analysis of Digital Images Using Copy-Move Forgery Detection

Alexander Sandoval Mesa Zhongming Wu

Abstract

In this report, we explore copy-move forgery detection on the CoMoFoD dataset, focusing on the small-size images, particularly those with translation-based forgeries. We develop and compare two approaches: a block-based method and a keypoint-based method. The block-based approach extracts features in the frequency domain and filters valid matches based on spatial consistency and shift vectors, achieving better performance than RANSAC. The keypoint-based approach combines SIFT and FREAK features, aligns their indices, and uses DBSCAN to cluster matching keypoints. Both methods demonstrate strong accuracy on translation and slightly rotated images but show limitations under large rotation angles and scaling. We conclude with a comparative analysis of the two methods' effectiveness.

1. Introduction

The proliferation of digital photography and image editing software has transformed how visual content is created, shared, and consumed. While these tools offer powerful creative possibilities, they also make it easier to alter images in ways that can mislead or deceive viewers. This has raised growing concerns across various fields, including media, security, and legal domains, where image authenticity is essential. As a result, the field of digital image forensics has emerged to address the need for reliable methods to assess the integrity of visual data. Motivated by the societal importance of trustworthy imagery, we explore techniques that support the detection of manipulated content through computational analysis.

1.1. Problem Statement

Among various types of image forgeries—such as splicing, removal, and retouching—copy-move forgery is one of the most common and challenging. It involves copying a region from an image and pasting it elsewhere within the same image, often concealing or duplicate objects. Because the duplicated region originates from the same image, it typically shares similar texture, lighting, and noise characteristics, making detection particularly difficult. The goal of copy-move forgery detection is to identify both the source and target regions of manipulation, regardless of transformations like translation, rotation, or scaling. This task becomes especially complex when forgeries are subtle, involve small objects, or are post-processed to mask traces of tampering. Our work focuses on detecting such manipulations in small-sized images, using both block-based and keypoint-based approaches.

2. Related Works

In this work, we focus on block-based and keypoint-based methods because they represent the most established and complementary approaches in CoMoFoD research.

Block-based methods are widely used in copy-move forgery detection for their ability to locate duplicated regions within an image. Fridrich et al. (2003) introduced a foundational approach using overlapping blocks and Discrete Cosine Transform (DCT) features, showing robustness to minor modifications. Popescu and Farid (2004) improved efficiency by applying Principal Component Analysis (PCA) to reduce feature dimensionality. Zhang et al. (2008) enhanced robustness to noise and rotation by combining block matching with Singular Value Decomposition (SVD). Ryu et al. (2010) further improved rotation and scale invariance by using Zernike moments for feature extraction.

Keypoint-based methods are widely used in copy-move forgery detection because they're fast and can handle changes like rotation, scaling, and compression. Unlike block-based techniques that scan the whole image, these methods focus only on distinctive points, making them more efficient and flexible.

One of the most well-known methods is SIFT (Lowe, 2004), which finds stable keypoints and describes them using gradient-based features. While it's accurate, SIFT can be slow due to its use of floating-point descriptors. To improve speed, binary descriptors like ORB (Rublee et al., 2011) and FREAK (Alahi et al., 2012) were developed. ORB uses fast keypoint detection and matching with Hamming distance, while FREAK mimics the human retina and compares pixel intensities to create compact binary descriptors.

Recent approaches often combine detectors like SIFT with fast descriptors like FREAK to get the best of both worlds—strong matching accuracy and good runtime performance, making them useful for real-world forensic applications.

3. Data Collection

The Copy-Move Forgery Detection (CoMoFoD) dataset was developed by researchers at the University of Zagreb in 2016 to support the benchmarking of copy-move forgery detection algorithms under diverse conditions. The dataset contains a total of 260 base (original) color images, each paired with 5 forged versions representing different types of manipulations: translation, rotation, scaling, distortion, and combinations of these transformations. Each forged image is accompanied by a binary ground-truth mask that indicates the exact regions involved in the copy-move operation. The image dimensions range from 512×512 to 3000×2000 pixels, though in our study we focus on a constrained subset of 200 small-sized images (typically 512×512) to reflect practical limitations. The dataset offers both straightforward and complex manipulation scenarios, allowing for controlled experiments on robustness to geometric and post-processing transformations.

4. Methodology Overview

To address the challenges of detecting copy-move forgeries, especially under geometric transformations, we explore two complementary approaches: a block-based method and a keypoint-based method. The block-based method operates in the frequency and spatial domains to identify duplicated regions, while the keypoint-based method leverages feature descriptors and clustering to localize forgeries. We focus primarily on detecting translation-only forgeries and extend our evaluation to images with slight rotation and scaling. Our expected outcome is that both methods will demonstrate high accuracy in detecting simple copy-move manipulations, with varying performance under more complex transformations. Detailed descriptions and evaluations of these methods are provided in the following sections.

4.1. Block-Based Method

We begin by focusing on translation-based copy-move forgeries, as they represent the most basic and common type of tampering. Our workflow is outlined below:

4.1.1 Block Extraction + DCT

We divide the grayscale image into overlapping blocks of size 8×8 pixels with a stride of 1 pixel. Each block undergoes a Discrete Cosine Transform (DCT) to capture frequency-domain information. DCT effectively separates image texture from intensity, which is helpful for detecting duplicated textures regardless of lighting conditions.

4.1.2 Zigzag Scanning + Quantization

DCT coefficients are reordered using zigzag scanning, prioritizing low-frequency components that are more stable and less affected by noise. We select the first 16 zigzag values, as they represent the most significant frequency features and strike a good balance between descriptor compactness and discriminability. Quantization (e.g., dividing by a constant) further reduces sensitivity to minor variations or compression artifacts.

4.1.3 Lexicographic Sorting

Feature vectors (DCT + position) are sorted lexicographically. This step ensures that similar features (e.g., duplicated blocks) appear close together in the array, significantly reducing computational complexity in the pairwise similarity comparison phase.

4.1.4 Similarity Calculation

We compute similarity between blocks by combining:

- Frequency similarity: the Euclidean distance between DCT vectors.
- Spatial constraint: the Euclidean distance between block positions.

Only block pairs with low frequency distance (i.e. tsimilarity <= 5) and sufficient spatial separation (i.e. tdistance >= 20) are considered.

This avoids false matches from neighboring similar blocks, which often occur naturally in textures (e.g., windows or bricks), and cannot be reliably identified as tampered regions.

This hybrid filter is more robust than using normalized cross-correlation (NCC) alone, which often fails under minor noise or JPEG compression.

4.1.5 Remove Noise & Boost Robustness

We use statistical shift vector filtering:

- For each valid block pair, compute its shift vector:

$\Delta x = x1 - x2$, $\Delta y = y1 - y2$

- Then count the frequency of each shift vector.

We assume that genuine copy-move operations generate a consistent shift vector, while noise results in randomly distributed shifts. By thresholding based on occurrence count, we suppress isolated matches and emphasize dominant, consistent manipulations. See Figure 1.

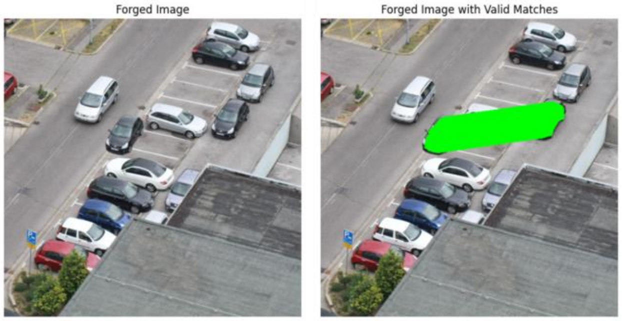


Figure 1. Filter Valid Matches with Shift Vector

Advantages of this method:

- Works well in repetitive structures.
- Does not require geometric model fitting (unlike RANSAC).
- Statistically principled and easy to implement.

Limitation:

- Performance is sensitive to the occurrence threshold, which must be tuned experimentally.

4.1.6 Model Evaluation

We first run the method with the initial setting (block_size = 8, stride =1) across all 40 translation-forged images and log performance metrics (F1, precision, recall, IoU).

Then, we loop over combinations of block_size and stride to identify the optimal parameter set with the highest average accuracy.

Table 1. Parameter and Description

Tuning Parameters	**Description**
block_size = [8, 16, 32]	patch size
stride = [1, 2, 4]	step size
quantization = 16	DCT compression
tsimilarity = 5	feature match threshold
tdistance = 20	min match distance
vector_limit = 20	min clone count

After evaluating different parameter combinations, we summarize the results in the pivot table below.

Table 2. Average F1 Score with Different Parameters

stride block_size	1	2	4
8	0.66123	0.255752	0.050657
16	0.583689	0.215806	0.018935
32	0.308199	0.104554	0.012119

Observations:

- Best setting: block_size = 8, stride = 1 yields the highest average F1-score (0.6612).
- Worst setting: block_size = 32, stride = 4 results in the lowest average F1-score (0.0121).

Trends:

- Increasing stride leads to a notable drop in F1-score.
- Larger block sizes generally reduce performance, especially when combined with higher stride values.

Best Model Analysis (block_size = 8, stride = 1): Under the optimal configuration, over 65% of the images achieve an F1-score above 0.7, indicating strong detection performance. Conversely, approximately 35% of the cases fall below an F1-score of 0.5, highlighting areas for improvement. See Figure 2.

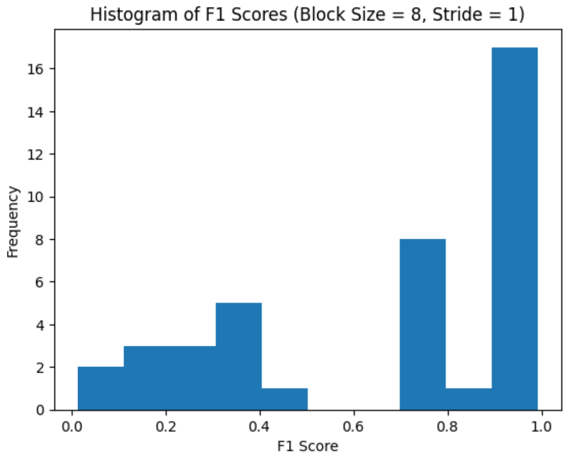


Figure 2. Histogram of F1 Score of the Best Model

The following visualizations illustrate the prediction outcomes for three representative cases, those with the highest, lowest, and median F1-scores. In each example, the ground truth mask is shown on the left, and the corresponding predicted mask is shown on the right. See Figure 3.

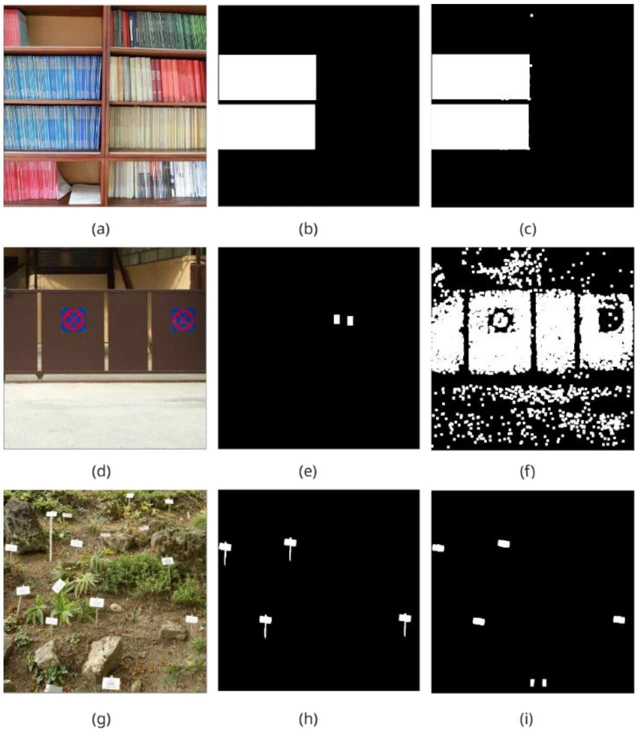


Figure 3. Highest, Worst and Median F1 Score

The image with the highest F1-score (0.9934) is image #24, presented in panels (a), (b), and (c). The image with the lowest F1-score (0.0130) is image #8, shown in panels (d), (e), and (f). The image representing the median F1-score (0.7529) is image #20, illustrated in panels (g), (h), and (i).

4.1.7 Generalization to Rotation & Scaling

Images with small rotation angles are still detected with reasonable accuracy, as local textures remain largely consistent. However, larger rotation angles disrupt the spatial alignment of DCT blocks, leading to mismatches and detection failure.

Block-based methods are ineffective under scaling transformations. Fixed-size blocks from the original and duplicated regions no longer correspond, causing DCT features and spatial locations to misalign. Even slight scale differences significantly hinder accurate matching due to the sensitivity of DCT to block size and shape.

This highlights a well-known limitation of block-based techniques: the lack of scale invariance. Refer to Figure 4 for visual results.

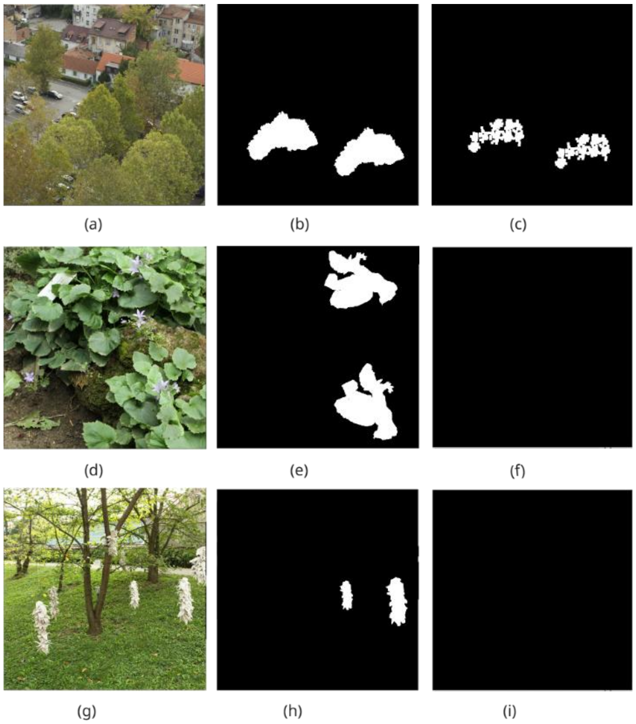


Figure 4. Performance on Rotated and Scaled Forgeries

In (a), (b), and (c), Image #48—subject to slight rotation—achieves a moderate F1-score of **0.6400**, indicating partial detection success.

In contrast, (d), (e), and (f) show Image #61 with significant rotation, resulting in a complete failure to detect forgery (**F1-score = 0**).

Similarly, (g), (h), and (i) present Image #102 with substantial scaling, also yielding an **F1-score of 0**, further confirming the limitations of block-based methods under scale transformations.

In contrast, keypoint-based approaches (e.g., SIFT) offer better resilience to geometric transformations.

4.2. Keypoint-Based Method

This approach leverages the strengths of local feature descriptors—specifically SIFT and FREAK—to identify duplicated regions within forged images. Unlike the dense block-based technique, this method focuses on distinctive and sparse keypoints (distinctive local features like corners, edges, textures), as the workflow is outlined below.

4.2.1 Image preprocessing

Before keypoint detection, each forged image in the CoMoFoD dataset undergoes several preprocessing steps to enhance texture detail and contrast. The preprocessing pipeline includes the following stages:

- **Resize:** Images are scaled to 512×512 pixels for consistent descriptor extraction.
- **Grayscale Conversion:** Converts RGB images to grayscale to reduce complexity and focus on intensity.
- **Laplacian Sharpening:** Enhances edges and textures by combining the Laplacian filter with the original image.
- **Brightness Enhancement:** Boosts visibility of darker features using intensity adjustment.
- **Histogram Equalization:** Normalizes intensity distribution to improve overall contrast and detail.

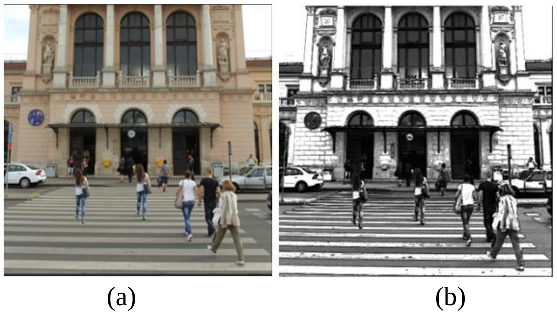


Figure 5. (a) Forged Image RGB and (b) preprocessed image with grayscale, Laplacian sharpening, brightness enhancement, and histogram equalization.

4.2.2 Keypoint Detection using SIFT

The Scale-Invariant Feature Transform (SIFT) is used to detect keypoints—distinctive regions in the image that remain stable under transformations like rotation, scaling, and illumination changes. For each keypoint, SIFT computes a 128-dimensional descriptor that captures the pattern of gradient orientations in the surrounding area, allowing for robust matching across different parts of the image.

To detect keypoints at multiple scales, SIFT constructs a Difference of Gaussians (DoG) pyramid by progressively blurring the image and subtracting adjacent blurred versions, as shown in Figure 6. This process highlights areas with strong intensity changes, such as edges, corners, and blobs, which are ideal locations for keypoint detection.

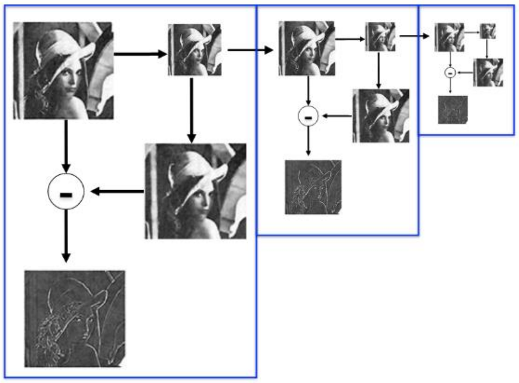


Figure 6. Difference of Gaussians (DoG) pyramid.

4.2.3 Descriptor Matching

To detect duplicated regions, keypoints were matched using a Brute-Force matcher with Hamming distance, which is well-suited for binary descriptors like FREAK (Fast Retina Keypoint). Each descriptor was compared with all others in the same image, and pairs with the lowest Hamming distances (threshold 15% of the maximum) were retained as potential forgery matches.

FREAK uses a retina-inspired sampling pattern, placing more sampling points near the keypoint center to capture fine details while still considering the broader context. Instead of using gradients, it performs intensity comparisons between point pairs, assigning a 1 if one pixel is brighter and 0 otherwise. The result is a compact binary descriptor (typically 512 bits) that describes the local image region.

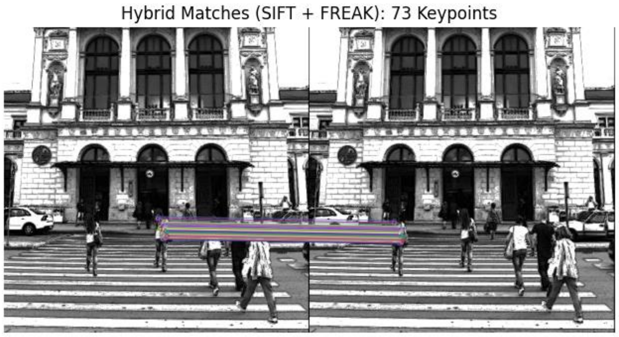


Figure 7. Matching keypoints using a duplicated image.

4.2.4 Clustering of Matches with DBSCAN

Keypoint matching alone yields isolated point correspondences that don't directly indicate the shape or location of forged regions. To group these into meaningful areas, we used DBSCAN (Density-Based Spatial Clustering of Applications with Noise)—an unsupervised algorithm that clusters points based on spatial proximity and filters out outliers. Each cluster is treated as a potential forged area, and bounding boxes are drawn around them to create the binary prediction mask.

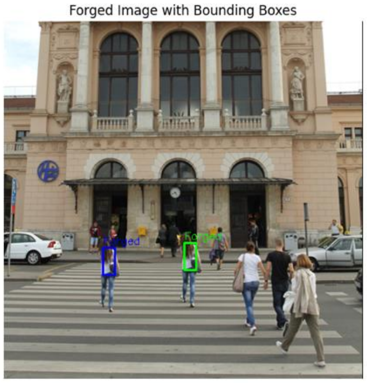


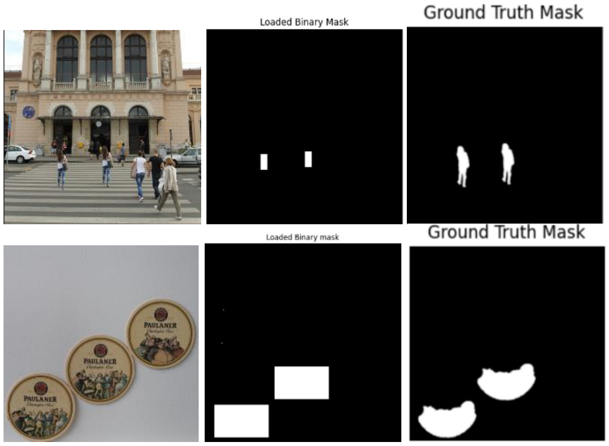
Figure 8. Forged regions with bounding boxes on the RGB forged image.

4.2.5 Binary Mask Generation

After clustering keypoint matches with DBSCAN, bounding boxes were drawn around each cluster to localize the duplicated regions. These were filled into a blank binary mask matching the size of the input image, where:

- Pixel value 1 (or 255) represents a forged region
- Pixel value 0 represents an authentic area

To improve coverage and compensate for gaps caused by sparse keypoints, morphological dilation was applied to slightly expand the detected regions. This refinement helps align the prediction more accurately with the ground truth during metric evaluation (e.g., F1-score, IoU).



(a) (b) (c)

Figure 9. (a) Forged Image, (b) Predicted image with Binary mask, and (c) Ground Truth mask.

4.2.6 Model evaluation

In this report, the F1 Score was selected as the primary metric to evaluate system performance, as it balances precision and recall, two critical aspects in detecting forgeries accurately and completely. IoU was also reported to assess the spatial alignment between the predicted and ground truth masks.

Translation

The proposed method showed strong performance in images. Out of 40 evaluated images, 33 achieved an F1 score above 0.7, indicating reliable detection of forged regions with a good balance between precision and recall. In terms of spatial accuracy, 23 images exceeded an IoU score of 0.7, showing that in over half the cases, the predicted masks were well-aligned with the ground truth.

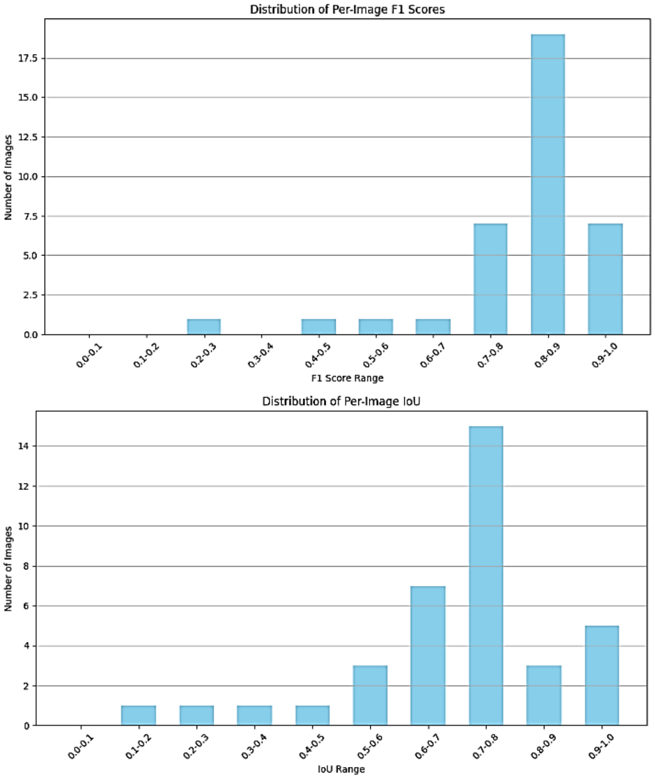


Figure 10. Distribution of F1 scores (top) and Intersection over Union (IoU) values (bottom) across all images.

Rotation

The system was specifically evaluated on forged images with rotated regions. Out of 40 images, only the 7 containing 180-degree rotated forgeries achieved meaningful detection results. Among these, 6 out of 7 had an F1 score above 0.7, and 6 out of 7 also had IoU scores above 0.7, with an average F1 of 0.706 and IoU of 0.613. In contrast, images with non-180° rotations (e.g., 30°, 60°, 120°) failed to yield detectable forgeries, indicating that the method is robust to 180° rotation but not generalizable to other angles.

Table 3. Performance on Forged Images with Rotated Regions

		F1 Score		
Rotation forged region	**Images**	0.4 – 0.5	0.7 – 0.8	0.8 – 0.9
R = 180 Degrees	7	1	4	2
R ≠ 180 Degrees	33	0	0	0
AVG Score		0.706		

		IoU		
Rotation forged region	**Images**	0.2 – 0.3	0.7 – 0.8	0.8 – 0.9
R = 180 Degrees	7	1	3	3
R ≠ 180 Degrees	33	0	0	0
AVG Score		0.613		

Scaling and distortion

Forged images involving scaling, affine distortions, or combinations of transformations (e.g., rotation with scaling or skewing) failed to produce valid detection results. These cases often resulted in empty or poorly aligned prediction masks, yielding unreliable F1 and IoU scores. This highlights a key limitation of the current approach: while it performs well under rigid transformations like pure translation and 180° rotation, it lacks robustness to complex geometric changes

Conclusion

The table above compares the performance of block-based and keypoint-based methods across 40 images containing pure translation forgeries.

Table 4. Average performance metrics

Metric	Block-Based	Keypoint-Based
Precision	0.619	0.862
Recall	0.912	0.820
F1 Score	0.661	0.810
IoU	0.575	0.703

This report compared two classic approaches for Copy-Move Forgery Detection (CMFD) on the CoMoFoD dataset: a block-based method using DCT features and a keypoint-based method combining SIFT and FREAK descriptors.

For images with pure translation, the keypoint-based approach demonstrated superior overall performance. It achieved significantly higher precision (0.862 vs. 0.619) and better F1 Score (0.810 vs. 0.661), indicating more balanced detection with fewer false positives. The block-based method, while achieving slightly higher recall (0.912), suffered from over-detection, leading to reduced localization accuracy, as reflected in a lower IoU score (0.575) compared to the keypoint-based approach (0.703).

Additionally, both methods showed reliable performance in cases involving 180° rotation, few images achieving F1 and IoU scores above 0.7. However, both methods struggled with scaling, affine distortions, and combinations of transformations, often failing to produce usable outputs. This highlights a key limitation of the current pipeline in handling complex geometric changes.

In summary, the keypoint-based method offers a more reliable and accurate solution for detecting forgeries under rigid transformations, making it better suited for practical forensic applications. Future work may focus on improving transformation invariance, especially for scale and affine distortions, by integrating geometric model fitting or deep learning-based feature extraction.

References

[1] Fridrich, J., Soukal, D., & Lukáš, J. (2003). Detection of copy-move forgery in digital images. *Proceedings of the Digital Forensic Research Workshop (DFRWS)*, Cleveland, OH, USA.

[2] Popescu, A. C., & Farid, H. (2004). Exposing digital forgeries by detecting duplicated image regions. *IEEE Transactions on Signal Processing*, 53(2), 758–767. https://doi.org/10.1109/TSP.2004.839932

[3] Zhang, J., Feng, Z., & Su, Y. (2008). A new approach for detecting copy-move forgery in digital images. *2008 IEEE International Conference on Communication Systems (ICCS)*, Guangzhou, China, 362–366. https://doi.org/10.1109/ICCS.2008.4737205

[4] Ryu, S. J., Lee, M. J., & Lee, H. K. (2010). Detection of copy-rotate-move forgery using Zernike moments. *Information Hiding: 12th International Conference, IH 2010*, Calgary, AB, Canada, 51–65. https://doi.org/10.1007/978-3-642-16435-4_5

[5] Alahi, A., Ortiz, R., & Vandergheynst, P. (2012). FREAK: Fast Retina Keypoint. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 510–517. https://doi.org/10.1109/CVPR.2012.6247715

[6] Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2), 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

[7] Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. *2011 International Conference on Computer Vision*, 2564–2571. https://doi.org/10.1109/ICCV.2011.6126544

Amerini, I., Ballan, L., Caldelli, R., Del Bimbo, A., & Serra, G. (2013). A SIFT-Based Forensic Method for Copy–Move Attack Detection and Transformation Recovery. *IEEE Transactions on Information Forensics and Security*, 6(3), 1099–1110. https://doi.org/10.1109/TIFS.2011.2129513

Course Project Workload Analysis Table

Tasks performed	Name of team member 1	Name of team member 2
Literature search	50%	50%
Algorithm derivation	50%	50%
Python programs	50%	50%
Report writing	55%	45%
PowerPoint presentation	45%	55%
Overall	(0.5+0.5+0.5+0.55+0.45)/5 = 50%	(0.5 + 0.5+0.5+0.45+0.55)/5 = 50%

Team Member #1: Alexander Sandoval Alexander Sandoval M.
Signature

Team Member #2: Zhongming Wu
Signature