Detection of Copy-Move Image Forgeries Using a SIFT-Based Approach

Serkan Hamdi GÜĞÜL and Göksun Güney KÜÇÜK

Boğaziçi University, Department of Electrical and Electronics Engineering
Report for EE475 Project

***Abstract*—With the widespread use of digital images in communication, social media, and forensic applications, ensuring the integrity and authenticity of visual content is critical. Copy-move forgery is a common manipulation technique where image regions are copied and pasted within the same image. This project explores a SIFT-based approach to detect such forgeries, leveraging its robustness to transformations like scaling and rotation. After SIFT method applied, a clusterinmg algorithm DBSCAN is applied and then with the help of a pre-trained segmentation model, DeepLabV3, forged patched are segmented. With this method we achieved to detect copy-move forgeries on simple images of the dataset CoMoFoD, with close performance to the modern methods.**

I. Introduction

With the widespread use of digital images in communication, social media, and forensic applications, the integrity and authenticity of visual content have become increasingly critical. Image forgery, specifically copy-move forgery, is a common technique used to manipulate images by copying and pasting regions within the same image to conceal or duplicate objects. This form of tampering is particularly challenging to detect due to its localized nature and the lack of external artifacts introduced during the process. This project aims to explore and implement a simple yet effective method for detecting copy- move forgeries in digital images. Copy-move forgery detection has been studied extensively using both traditional and deep learning-based approaches. One particular group of methods rely on feature extraction techniques such as Scale-Invariant Feature Transform (SIFT) [1]. and Speeded-Up Robust Features (SURF) [2] to identify duplicated regions. In contrast, deep learning methods leverage convolutional neural networks (CNNs) [3] and other advanced architectures to detect forgeries with higher accuracy and robustness.

II. Overview of Classical Copy-Move Forgery Detection Methods

Copy-move forgery detection methods can be broadly categorized into five groups based on their feature extraction techniques. These groups are moments, dimensionality reduction, intensity, frequency and keypoint-based methods. Table below table some of the prominent methods for each group. Keypoint- based methods, particularly SIFT and SURF, are widely used due to their robustness against transformations such as rotation, scaling, and illumination changes.

Group	Methods	Feature-length [1]
Moments	Blur [20]	24
	Hu [26]	5
	Zernike [24]	12
Dimensionality reduction	PCA [23]	–
	SVD [13]	–
	KPCA [4]	192
Intensity	Luo [19]	7
	Bravo [7]	4
	Lin [18]	9
	Circle [27]	8
Frequency	DCT [10]	256
	DWT [4]	256
	FMT [6]	45
Keypoint	Sift [11], [22], [3]	128
	Surf [36], [37]	64

TABLE I
Groping of copy-move forgery detection methods

A. SURF

The Speeded-Up Robust Features (SURF) [2] algorithm, introduced by Bay et al. in 2006, is a keypoint-based method designed for detecting and describing local features in images. It was developed as a computationally faster alternative to the Scale-Invariant Feature Transform (SIFT), while maintaining comparable accuracy and robustness. SURF is widely used in applications like object recognition, image registration, and forgery detection due to its ability to handle various transformations, such as scaling, rotation, and partial occlusions. In the context of copy-move forgery detection [4], SURF is used to extract keypoints from an image and match them to find duplicated regions. Its robustness and speed make it suitable for detecting forgeries in large images or datasets, where computational efficiency is critical. Although SURF is faster than SIFT, its performance can sometimes be less accurate, particularly in handling extreme transformations. Nevertheless, it remains a powerful tool for image analysis tasks, including forgery detection.

B. SIFT

The Scale-Invariant Feature Transform (SIFT), proposed by David Lowe in 1999 and refined in 2004, is a keypoint detection and description algorithm widely used in computer vision. It is designed to identify and describe local features in an image, making it robust to transformations such as

scale, rotation, and illumination changes. SIFT is particularly useful for tasks like object recognition, image matching, and forgery detection due to its accuracy and robustness. The SIFT algorithm involves four main steps:

- Scale-space representation using Gaussian kernels.
- Keypoint localization via extrema detection in Difference of Gaussians.
- Orientation assignment based on local gradients.
- Descriptor generation by forming histograms of gradient orientations.

1) Scale-Space Representation: To ensure scale invariance, SIFT identifies keypoints in different scales using a scale-space representation. The scale space of an image I(x,y)I(x, y)I(x,y) is generated by convolving the image with a Gaussian kernel:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

The Difference of Gaussians (DoG) is computed by subtracting Gaussian-blurred images at two nearby scales:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$

This efficiently approximates the Laplacian of Gaussian (LoG) to detect scale-invariant keypoints. Potential keypoints are identified as local extrema in the DoG images. A point is considered a keypoint if it is the maximum or minimum among its 26 neighbors (8 in the current scale, 9 each in the scales above and below).

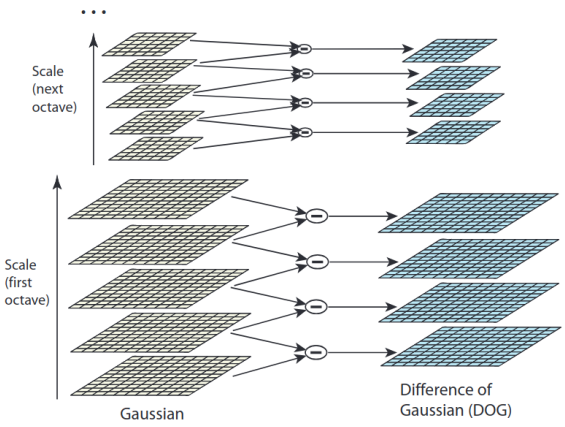


Fig. 1. For each octave of scale space, the initial image is repeatedly convolved with Gaussians to produce the set of scale space images shown on the left. Adjacent Gaussian images are subtracted to produce the difference-of-Gaussian images on the right. After each octave, the Gaussian image is down-sampled by a factor of 2, and the process repeated.

2) Key Point Localization: Key points are identified as local extrema in the DoG function. For a pixel to be a key point, it must be a maximum or minimum compared to its 26 neighbors in the 3D space (8 neighbors in the current scale and 9 in each of the scales above and below). To refine the key point location, a Taylor expansion of D around the candidate key point is performed:

$$\widehat{\mathbf{x}} = -\frac{\partial^2 D^{-1}}{\partial \mathbf{x}^2} \frac{\partial D}{\partial \mathbf{x}}$$

he extrema is accepted if the intensity at x is above a contrast threshold and not on an edge (based on the ratio of eigenvalues of the Hessian matrix).

3) Orientation Assignment: Each key point is assigned an orientation based on the gradient magnitude and direction within its local neighborhood. The gradient is computed as:

$$m(x, y) = \begin{array}{r} \sqrt{(L(x+1, y) - L(x-1, y))^2} \\ +(L(x, y+1) - L(x, y-1))^2 \end{array}$$

$$\theta(x, y) = \tan^{-1}\left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}\right)$$

A histogram of orientations is constructed, weighted by $m(x, y)$ and the dominant orientation θ_{max} is selected. Additional orientations are added for other significant peaks to ensure robustness.The region around each key point is divided into n×n subregions (typically 4×4). Within each subregion, the gradient orientations are binned into 8 directions. The descriptor for the key point is formed by concatenating these histograms. This result in a feature vector with 128 entries.

III. Deep Learning Methods and Related Work

A impactful approach to copy-move detection using deep learning methods made my Ouyang et al. [3]. They present a novel approach to copy-move forgery detection (CMFD) using convolutional neural networks (CNNs). Traditional CMFD methods are typically divided into key-point-based and block-based techniques. Key-point-based methods, such as those using Scale-Invariant Feature Transform (SIFT) and Speeded-Up Robust Features (SURF), are efficient at detecting duplications under transformations like rotation, scaling, and translation. However, they struggle with regions that lack visual structure or distinctive key points. Block-based methods, on the other hand, divide an image into overlapping blocks and match features, but they can be computationally expensive and less effective under complex distortions. The proposed CNN-based method addresses these challenges by leveraging pre-trained models from large datasets, such as ImageNet, to reduce the need for extensive new training data. Using a fine-tuning process, the network adjusts its structure to accommodate smaller CMFD-specific datasets. This enables the model to detect tampered regions with greater accuracy while maintaining computational efficiency. The paper evaluates its method on various datasets, including those generated by simple copy-move operations and real-world forgeries. Results show that the CNN approach performs.

A. Dense-InspectionNet

The Dense-InceptionNet proposed by Zhong and Pun is an end-to-end deep neural network architecture designed for image copy-move forgery detection. It comprises three main components: the Pyramid Feature Extractor (PFE), the Feature

Correlation Matching (FCM) module, and the Hierarchical Post-Processing (HPP) module. The PFE module extracts multi-scale and multi-dimensional dense features using a combination of DenseNet and Inception architectures. The FCM module autonomously learns feature correlations to generate matching maps that identify potential forgery regions. Finally, the HPP module refines these matching maps through cross-entropy computations and backpropagation, improving the model's accuracy. This architecture effectively addresses challenges in detecting untrained forgery classes and outperforms several state-of-the-art methods in efficiency and accuracy.

B. DOG-GAN

The Dual-Order Attentive Generative Adversarial Network (DOA-GAN), proposed by Islam et al., is designed for image copy-move forgery detection and localization. The architecture consists of a generator and a discriminator. The generator employs a dual-order attention module that calculates two attention maps: the 1st-order attention map captures location-aware features to identify manipulated regions, while the 2nd-order attention map explores inter-patch dependencies for finer discrimination. These attention maps are fused with features extracted through atrous spatial pyramid pooling (ASPP) blocks, enabling robust detection of source and target regions. The discriminator verifies the predicted mask against the ground truth to improve accuracy through adversarial training. The framework achieves state-of-the-art performance in detecting copy-move forgeries across multiple benchmark datasets.

IV. Matching, Clustering and Segmentation in Our Project

A. FLANN: Fast Library for Approximate Nearest Neighbors

FLANN (Fast Library for Approximate Nearest Neighbors) is a library designed to perform fast and efficient approximate nearest neighbor searches in high-dimensional spaces. This technique is particularly useful in applications such as image retrieval, object recognition, and 3D reconstruction, where the data points have many dimensions and exact nearest neighbor search can be computationally expensive.

Key Features of FLANN

- **Approximate Search:** FLANN focuses on approximate nearest neighbors instead of exact matches, significantly speeding up the search process while maintaining high accuracy.
- **Automatic Algorithm Selection:** It automatically selects the best nearest neighbor algorithm and optimizes parameters for the given dataset, based on benchmarking.
- **Scalability:** The library is scalable and can handle large datasets with millions of data points.
- **Ease of Use:** FLANN provides an easy-to-use interface with support for multiple programming languages, including C++ and Python.

FLANN implements several algorithms for nearest neighbor search, including:

- **KD-Trees:** A space-partitioning data structure that organizes points in a k-dimensional space, allowing efficient search queries.
- **Hierarchical Clustering Trees:** A tree-based structure where data points are grouped hierarchically, enabling efficient approximate search.
- **Other Approximation Methods:** FLANN includes additional techniques for balancing speed and accuracy, such as random projection trees.

When using FLANN, the user specifies a dataset and a query set. FLANN benchmarks its internal algorithms on the dataset and selects the one that provides the best trade-off between speed and accuracy. FLANN is used in the following areas:

- **Computer Vision:** FLANN is commonly used in applications like object recognition, feature matching, and structure-from-motion pipelines.
- **Image Retrieval:** It helps retrieve similar images from large datasets based on extracted features.
- **Machine Learning:** FLANN is used to speed up k-Nearest Neighbors (k-NN) classification tasks.
- **3D Reconstruction:** Efficiently finds correspondences between 3D points in large-scale datasets.

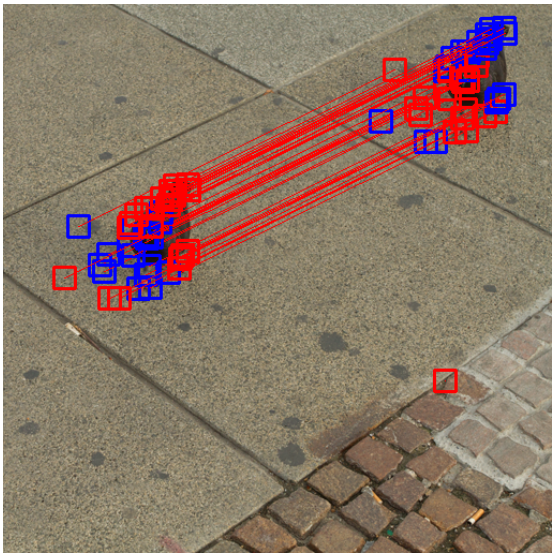


Fig. 2. An example of FLANN matching after applying SIFT to forged image

B. DBSCAN: Density-Based Spatial Clustering of Applications with Noise

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a clustering algorithm that identifies dense regions in data separated by sparser regions. In the context of copy-move forgery detection, DBSCAN can be applied to cluster SIFT keypoints based on their spatial proximity.

Given two parameters, ε (the radius of a neighborhood) and `MinPts` (the minimum number of points to form a dense region), DBSCAN classifies points as: 1. *Core points*, which have at least `MinPts` neighbors within a distance ε. 2. *Border points*, which are within the ε-neighborhood of a core point

but do not satisfy the `MinPts` condition themselves. 3. *Noise points*, which are neither core nor border points.

For SIFT keypoints, DBSCAN can group nearby keypoints into clusters, representing areas of dense activity that are likely candidates for copy-move forgery regions. This clustering step is critical for isolating potential regions of forgery before further analysis.

In figure 3, there is an explanation for the algorithm.

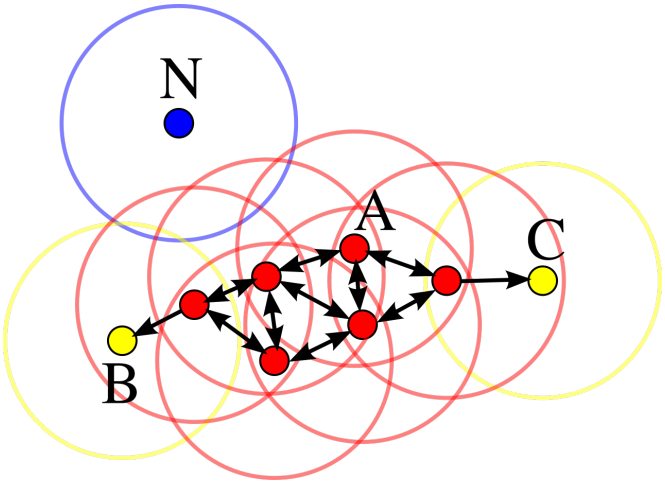


Fig. 3. In this diagram, minPts = 4. Point A and the other red points are core points, because the area surrounding these points in an radius contain at least 4 points (including the point itself). Because they are all reachable from one another, they form a single cluster. Points B and C are not core points, but are reachable from A (via other core points) and thus belong to the cluster as well. Point N is a noise point that is neither a core point nor directly-reachable.

C. DeepLabv3

DeepLabv3 is a state-of-the-art image segmentation model that leverages atrous convolution and Atrous Spatial Pyramid Pooling (ASPP) to achieve accurate semantic segmentation by capturing multi-scale contextual information.

Atrous convolution increases the receptive field of a convolutional filter without increasing the number of parameters or reducing the spatial resolution of the input. This is achieved by introducing a dilation rate, which determines the spacing between elements of the convolutional kernel. A dilation rate of 1 corresponds to standard convolution, while larger dilation rates expand the receptive field by skipping pixels between kernel elements. This allows DeepLabv3 to capture features from larger regions of the image while preserving fine details.

To handle objects at different scales, DeepLabv3 incorporates Atrous Spatial Pyramid Pooling (ASPP), a technique that applies multiple atrous convolutions with varying dilation rates in parallel. Each atrous convolution captures features from a specific receptive field, while global average pooling incorporates global context. The outputs from these layers are concatenated and passed through a final convolutional layer to produce the segmentation map. This multi-scale approach enables DeepLabv3 to effectively segment objects of varying sizes within complex scenes.

By combining atrous convolution and ASPP, DeepLabv3 balances high-resolution feature extraction with the ability to capture broad contextual information, making it highly effective for semantic segmentation tasks.

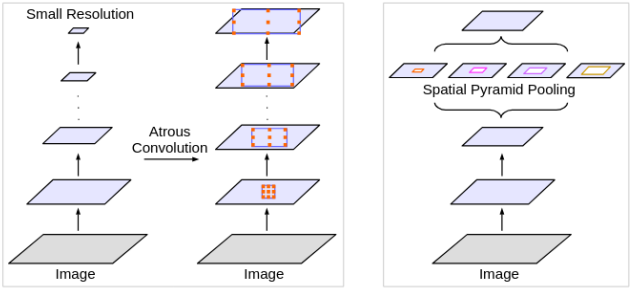


Fig. 4. Atrous Convolution and ASSP

V. Our Approach

As explained in the previous chapters, there are bunch of methods to tackle the copy-move detection problem. Our approach is presented in below figure.

We decided to use a SIFT- based (Scale-Invariant Feature Transform) aproach in our algorithm because it is highly effective at detecting distinctive key points in an image that are invariant to scale, rotation, and partial affine transformations. This makes it ideal for identifying copy-move forgeries, where a region of the image is copied and pasted elsewhere. There are many research about the performance and the methodology of using SIFT [6,7]. Our algorithm use a more simpler approach for know, we will implement more advanced approaches later. Further, we are using the CoMoFoD (Image Database for Copy-Move Forgery Detection) dataset. [8]. Our algorithm is explained in detail:

The algorithm begins by converting the input image into grayscale, as SIFT operates on intensity values rather than color. We then apply SIFT to detect key points, which are distinctive features in the image, and compute their descriptors, which are numerical representations of the local image structure. Next, we use a FLANN-based matcher (Fast Library for Approximate Nearest Neighbors) [9] to match the descriptors within the same image. This step helps us identify potential duplicate regions. To improve the reliability of the matches, we apply Lowe's ratio test, which compares the second-best match to the third- best match. We retain only the matches that meet a set threshold, discarding others to minimize false positives.

Even after selecting good matches, there may still be false alarms due to real-life repetitions (like stripes on a road). To address this, we apply DBSCAN (Density-Based Spatial Clustering of Applications with Noise). For a matched region to be labeled as a copy-move forgery, there must be an accumulation of matches for different keypoints. In this step, we determine the allowable size of a cluster and the minimum required matches to label a patch as copy-moved.

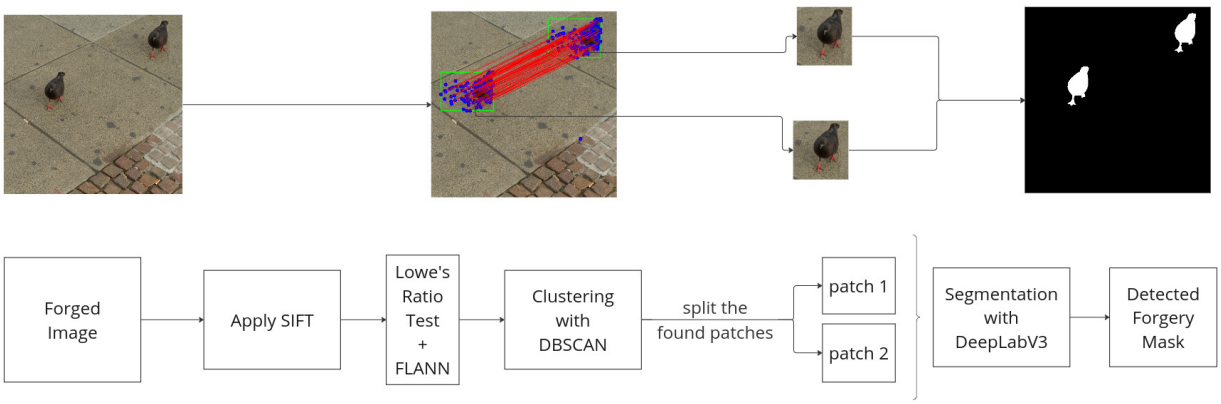


Fig. 5. Flow of Our Algorithm for Copy-Move Forgery Detection

Once we identify valid matches, we extract the coordinates of the matching key points and draw rectangles around these points to highlight the potential duplicate regions. We also draw lines between the matched key points to visually connect the regions that may have been copied.

Furthermore, DeepLabv3 was used for precise segmentation of forged regions by leveraging its deep learning-based semantic segmentation capabilities. It employs atrous convolution to capture multi-scale context. We crop the detected regions that we found after applying SIFT and DBSCAN. Then we pass those patches to model and we get segmentation maps. We place those segmentation maps of the patches into their original places. Thus, we get the segmentation map of the forged image. We implemented our algorithm to the all the 200 images in the CoMoFoD. Below there are some examples of good performing results with the poor ones later. For better understanding of the reader the mask and original images are also included.

VI. Evaluation

There are many metrics to test the performance of the detection models. In our project we used the following ones:

- **Precision** (P): Measures the accuracy of positive predictions.

$$P = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

- **Recall** (R): Measures the ability to find all relevant positive instances.

$$R = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

- **F1 Score** (F_1): Harmonic mean of precision and recall.

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R}$$

- **Accuracy** (A): Measures overall correctness of the model.

$$A = \frac{\text{True Positives (TP)} + \text{True Negatives (TN)}}{\text{Total Instances}}$$

- **Goodness Score** (G): A measure based on set comparison.

$$G = \frac{D \cap G}{D \cup G}$$

As we mentioned earlier, there are many methods for copy-move forgery detection. Here we will compare our results with other studies on CoMoFoD dataset. Be awere that we only evaluated our model on CoMoFoD's simple images, ie non-processed ones. So we have a higher chance to get good results. However, we want to show that our simple method achieve similar results to SOTA models when tested on those simple images:

TABLE II
COMPARISON ON COMOFOD DATASET
*OUR METHOD IS ONLY EVALUATED ON THE NON-PROCESSED IMAGES OF COMOFOD

Methods	**Year**	**Precision**	**Recall**	**F1 Score**
Block-ZM	2010	51.72	20.87	29.74
DCT-Match	2012	50.48	29.77	37.46
Adaptive-Seg	2015	65.66	43.37	52.24
DenseField	2015	**80.34**	20.10	32.15
BusterNet	2018	53.20	57.41	55.22
DOA-GAN	2019	60.38	**65.98**	**63.05**
Our Method*	2025	52.17	46.99	42.91

TABLE III
OUR EVALUATION METRICS

Metric	**Value**
Accuracy	0.9576
Precision	0.5217
Recall	0.4699
F-1 Score	0.4291
Goodness-Score	0.3218

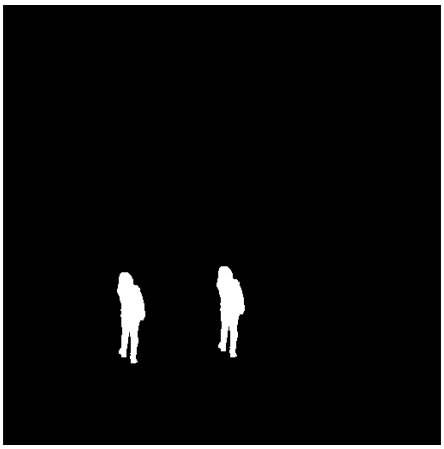
VII. RESULTS

We applied our algorithm to the **200 images** in the CoMoFoD dataset. Here are some good results:
Comparison of results: (a) Original image, (b) Forged image, (c) SIFT application, (d) Segmentation, (e) Ground truth.

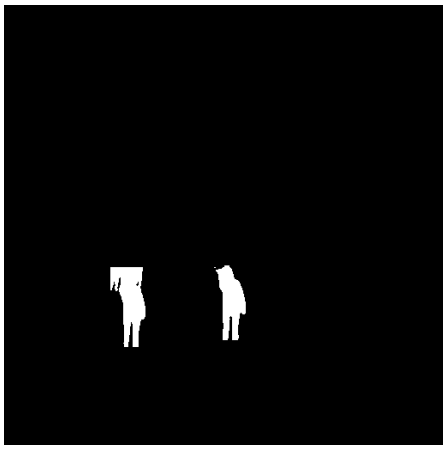
Original

Fake

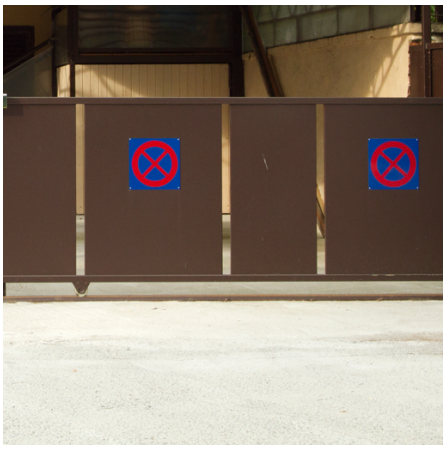
SIFT-applied



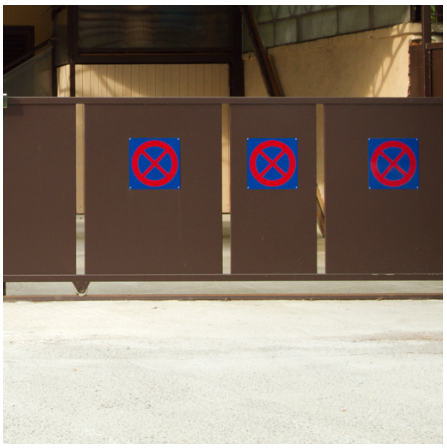
Ground-truth map



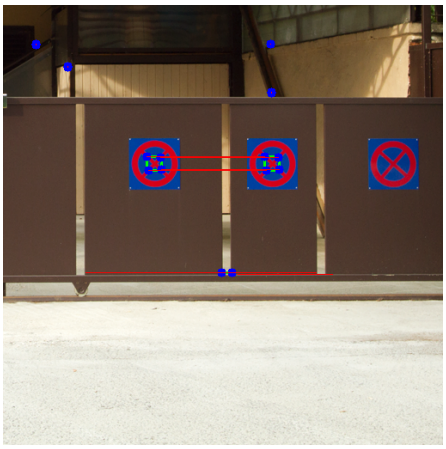
Ours segmentation map



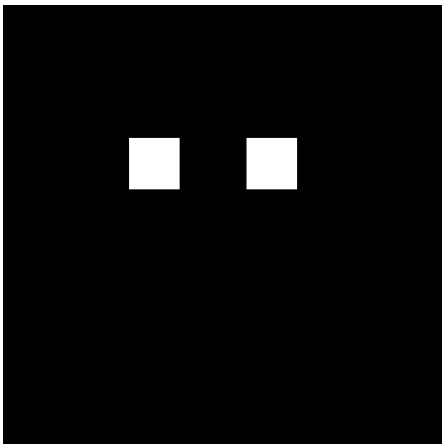
Original



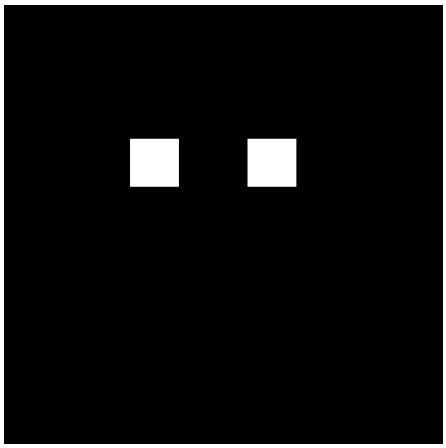
Fake



SIFT-applied



Ground-truth map



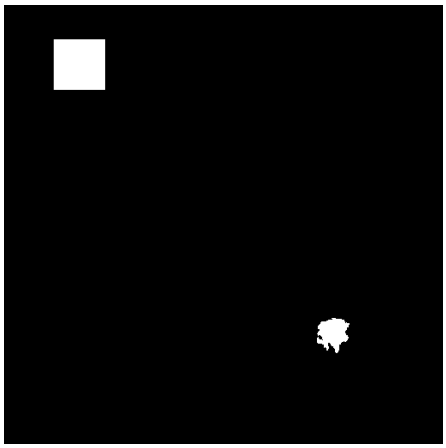
Ours segmentation map

Original

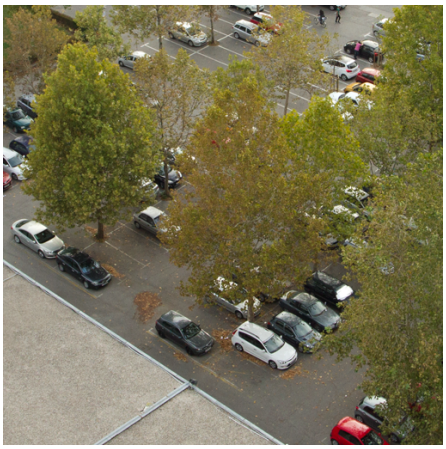
Fake

SIFT-applied

Ground-truth map

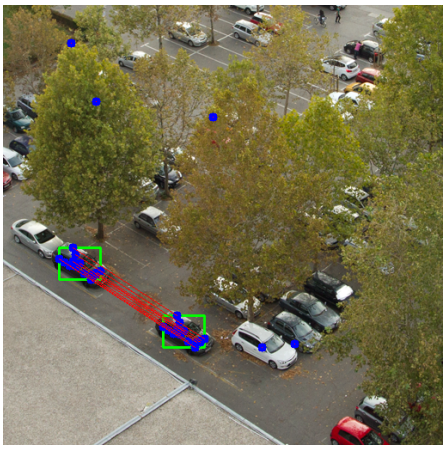


Ours segmentation map



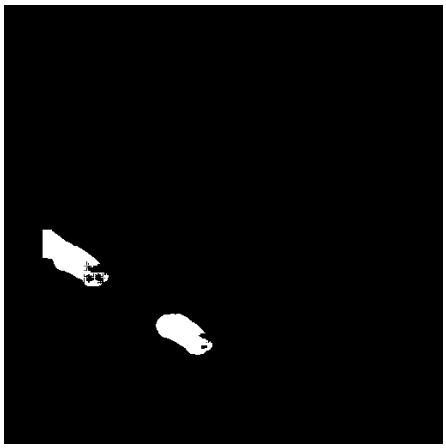
Original

Fake



SIFT-applied

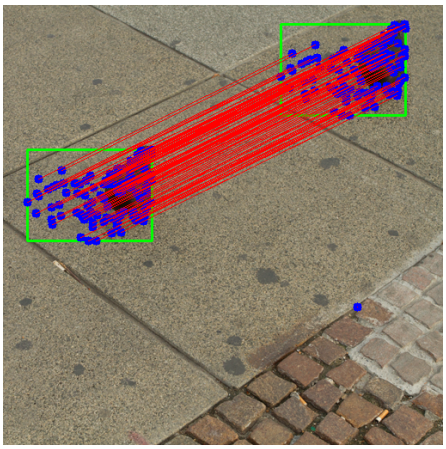
Ground-truth map



Ours segmentation map

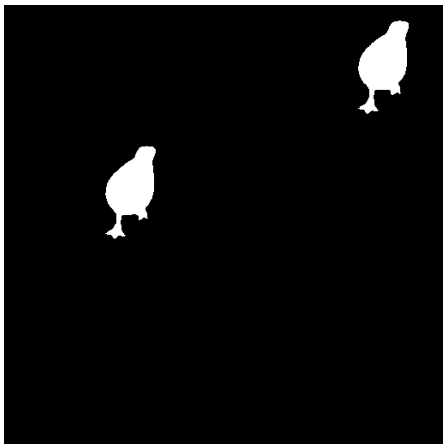
Original

Fake



SIFT-applied

Ground-truth map



Ours segmentation map

Original

Fake

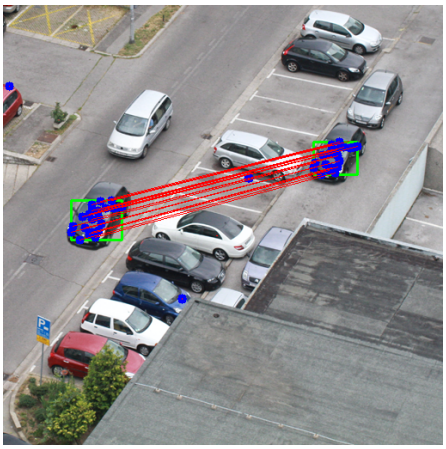
SIFT-applied

Ground-truth map

Ours segmentation map

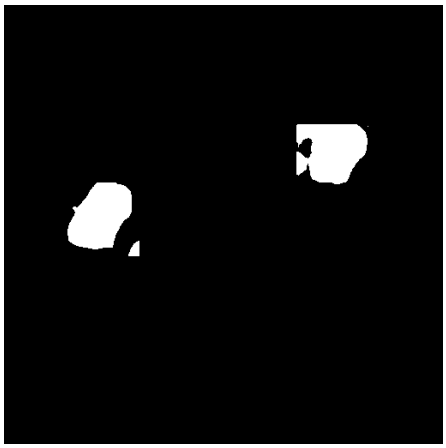
Original

Fake



SIFT-applied

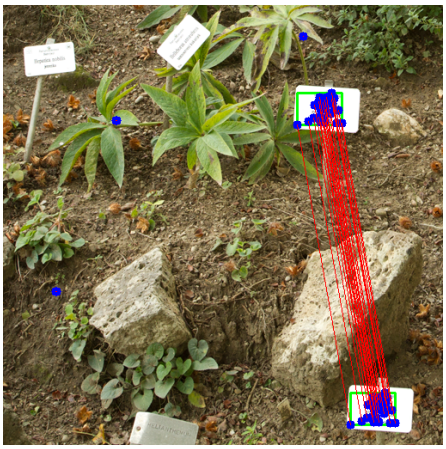
Ground-truth map



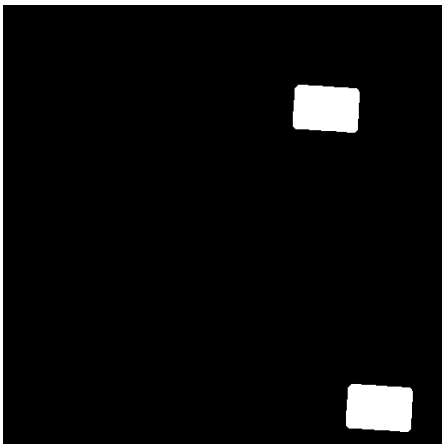
Ours segmentation map

Original

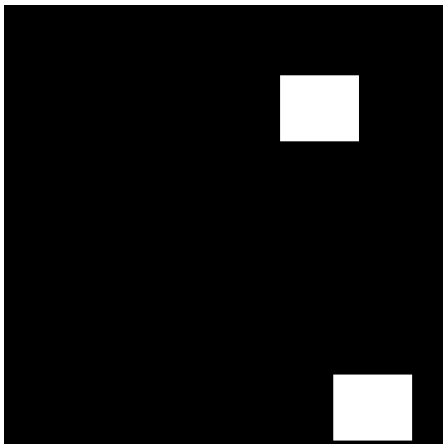
Fake



SIFT-applied



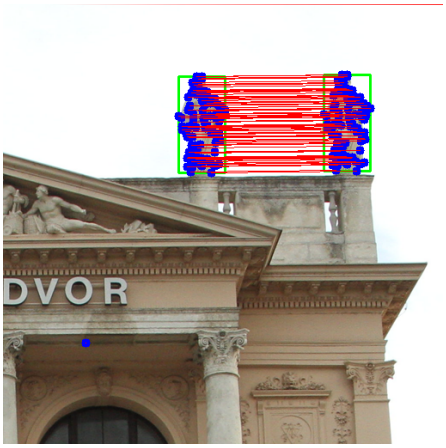
Ground-truth map



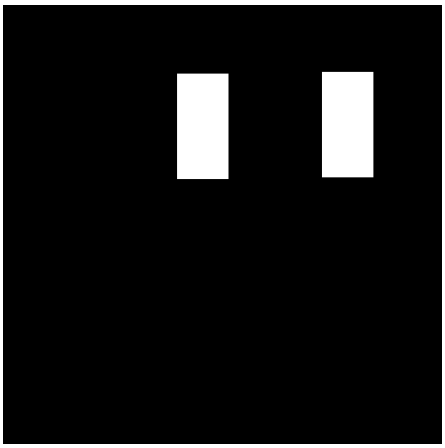
Ours segmentation map

Original

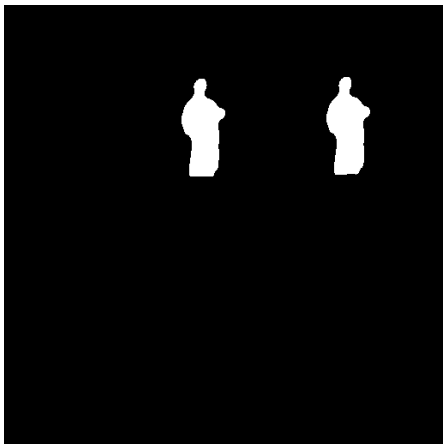
Fake



SIFT-applied



Ground-truth map



Ours segmentation map

VIII. Conclusion

In this project, we explored a method for detecting copy-move forgeries in digital images by combining traditional feature-based techniques with modern segmentation approaches. Our algorithm utilizes the SIFT (Scale-Invariant Feature Transform) method for robust feature extraction, followed by DBSCAN clustering to group spatially close keypoints that could indicate duplicated regions. To refine the detection and achieve precise localization, we incorporated the DeepLabv3 segmentation model, leveraging its advanced atrous convolution and ASPP techniques for multi-scale context capture.

The proposed approach was applied to the CoMoFoD dataset, demonstrating its ability to detect copy-move forgeries in relatively simple scenarios with performance comparable to more advanced methods. The use of SIFT ensured robustness against transformations like scaling and rotation, while DBSCAN helped isolate meaningful clusters from potential noise. The integration of DeepLabv3 allowed us to generate detailed segmentation maps for suspected forgery regions, further enhancing the accuracy of detection.

While the algorithm performed well on simpler examples, challenges remain in handling more complex images with high texture diversity, illumination changes, or overlapping objects. Future work can focus on improving the robustness of the algorithm by exploring advanced feature matching techniques, optimizing DBSCAN parameters, and incorporating additional deep learning-based methods for more complex forgery scenarios.

This project highlights the effectiveness of combining traditional feature-based methods with modern deep learning approaches to tackle the problem of copy-move forgery detection. It sets a foundation for further exploration and development of hybrid techniques that leverage the strengths of both paradigms to enhance the reliability and accuracy of image forensics tools.

References

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004, Springer.

[2] H. Bay, "Surf: Speeded up robust features," *Computer Vision—ECCV*, 2006.

[3] J. Ouyang et al., "Copy-move forgery detection based on deep learning," *10th Int. Congress on Image and Signal Processing*, 2017.

[4] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, vol. 96, no. 34, pp. 226–231, 1996.

[5] L.-C. Chen, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[6] J.-L. Zhong and C.-M. Pun, "An end-to-end dense-inceptionnet for image copy-move forgery detection," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2134–2146, 2019.

[7] V. K. Singh, A. Singh, and S. Jain, "Zernike moments-based rotation invariant property for fast copy-rotate-paste image forgery detection," *International Journal of Computer Applications*, vol. 89, no. 16, pp. 26–32, 2014.

[8] N. Jadhav, S. Kumar, and H. Singh, "DCT-based methods for robust copy-move forgery detection," *International Journal of Advanced Research in Computer Science and Management Studies*, vol. 4, no. 2, pp. 201–203, 2016.

[9] G. Muhammad and S. Kumar, "Adaptive segmentation for efficient forgery detection using histogram-based techniques," *Journal of Image and Video Processing*, 2016.

[10] S. Roth and M. Black, "Dense field estimation for identifying geometric transformations in forged images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008.

[11] J.-L. Zhong and C.-M. Pun, "An end-to-end dense-inceptionnet for image copy-move forgery detection," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2134–2146, 2019.

[12] A. Islam, X. Zhang, Y. Zhang, and Z. Liang, "DOA-GAN: Dual-order attentive generative adversarial network for image copy-move forgery detection and localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.