Splicing Forgery Detection in Digital Images

***Abstract*—Never has digital image manipulation been more easily and persuasively allowed due to the growing accessibility of sophisticated photo-editing tools. This has made the process of determining the authenticity of digital images a very important exercise in media forensics, in law enforcement, in investigating insurance cases, and documenting scientific findings. The project describes a complete solution of Image Tampering Detection with particular attention paid to Forgery Localization, which is not only to obtain responses to the question whether an image is tampered with or not, but also to locate the places where the manipulation has taken place.**

The suggested system combines the traditional image forensic methods such as Error Level Analysis (ELA), block-based matching and feature-based comparison to identify alterations made in case of copy-move forgeries, as well as splicing forgeries. To facilitate the extraction of dataset, preprocessing, feature generation, thresholding, morphological refinement and visualization of forged regions a structured pipeline was adopted. Moreover, the parameter tuning experiments were performed in order to investigate the systematic analysis of the impact of ELA quality, binary thresholding and morphological kernel sizes on the accuracy and clarity of results of detection. It is proven through the results of the experiment that the created pipeline successfully identifies manipulated areas and generates informative heatmaps that help to observe possible forgeries. By and large, the system demonstrates the future possibilities of the traditional image forensics techniques in the real-world tampering-detection use cases and preconditions the further advancement of the techniques by the means of machine learning and deep visual representation.

***Index Terms*—Image forensics, image tampering detection, forgery localization, copy-move forgery, image splicing, Error Level Analysis (ELA), block matching, feature matching, morphological processing, binary thresholding, compression artifacts, digital image authentication, Columbia image splicing dataset, heatmap visualization, image manipulation detection.**

I. Introduction

Digital images are very important in the communication, documentation and decision making in a very broad field including journalism, surveillance, medical imaging and court cases. Nevertheless, due to the extensive accessibility of powerful image editing software like Photoshop, Gimp, and AI-based generative models, the manipulation of images has become simpler, smoother and usually unnoticed to the naked eye. These developments have become a major problem with regard to media integrity since, through manipulated images, a viewer can be mislead, facts can be embellished, or fraudulent dealings can be perpetuated. This has led to the creation of credible and automated image tampering detection algorithms, an indispensable research field in digital forensics.

Common techniques used in image tampering include copy-move forgery, which involves duplicating a section of an image and pasting it into another part of the image to either hide or imitate objects in an image and splicing, which involves using contents of one image to create another by combining them. To detect these manipulations, it is necessary to detect changes in the natural features of images, including compression levels, noise structure, uniformity of textures or symmetry of structures, which occur in the process of editing.

The project is a project on Forgery Localization, which is a more difficult type of image forensics, which does not just attempt to establish whether an image has been manipulated or not, but the locations of the manipulated locations. In order to do it, we modeled a multi-stage processing pipeline with the classical techniques of forensics. The system uses the Error Level Analysis (ELA) to reveal differences in JPEG compression, block-based matching to detect duplicated patterns which mean that someone has copied an image, as well as feature matching to detect geometric discrepancies between the original and tampered output. Further processing to refine candidate areas and improve the eventual visualization is done by other measures like binary thresholding and morphological processing.

The data workflow starts with extraction and preliminary processing of the dataset with the help of Columbia Uncompressed Image Splicing Detection Dataset and further analysis of the inconsistency of the features in a systematic way. Simulation experiments were performed using different ELA qualities, threshold values, and morphological kernel sizes in order to have a better insight into the impact of algorithmic parameters. The tests give a more insight into the interaction of each parameter on the sensitivity and accuracy in the detection process.

Combining formal experimentation with classical analysis, this project proves that there is a successful and understandable way of detecting image tampering. The results are not only a part of the general body of retaining the integrity of digital images but also indicate a chance of future research in deep learning, noise-sensitive descriptors, and hybrid forensic systems.

II. Methodology

A. Theory

Error Level Analysis (ELA): Error Level Analysis is applied to expose the compression anomalies that are usually apparent in the distorted areas of an image. Natural areas undergo consistent JPEG degradation when the image is re-saved at the same quality, but spliced or edited areas have varying levels of error. In the adopted approach, the original input picture is re-saved at JPEG quality of $Q = 80$ and a pixel-by-pixel difference between the original image and resaved picture is calculated. The ELA heatmap image is

normalized into the range [0,255] to indicate the possible spliced areas as the difference image.

Ground Truth Mask Extraction: The Columbia dataset has ground truth masks, which represent manipulated regions with the green color channel. In order to extract these masks, the algorithm reads in the mask image in BGR format and weighs up each pixel with respect to the channel intensity relationships. A pixel is considered spliced when the green intensity of the pixel is larger than the red and the blue intensities and larger than a small value. This generates a pure binary ground truth mask to evaluate the performance of detetction.

Threshold-Based Splicing Localization: The heatmap of manipulated regions is predicted by converting ELA heatmap to grayscale. A world-wide threshold of T = 25 is used to isolate high error pixels. Areas that have high ELA scores have higher chances to represent tampered content as these pixels have increased inconsistencies in the recompression stage. The binary mask resulting gives an approximate indication of manipulated regions.

Morphological Refinement: Though thresholding can be used to recognize large regions of manipulation, it can also create noise or create gaps. Morphological operations are used one after another to refine the mask. An elliptical structuring component of size, K = 3, is employed. Morphological opening eliminates tiny noisy elements, whereas morphological closing fills tiny holes within identified areas. This gives a smoother and more precise predicted mask.

Contour-Based Visualization: To be inspected qualitatively, the refined prediction mask is contours extracted and drawn on the original RGB image. This is only used to compare the detected regions visually to the actual manipulated regions although not used to do any metric computation.

Dataset Handling and Mask Selection: The situation with the Columbia dataset is that each spliced image has several filenames of ground truth mask images with varying suffixes. The algorithm creates every possible version of a mask file name, and picks the first file existing. The image is skipped in case of a lack of a valid mask.

Evaluation Metrics: Both predicted and ground truth masks are reduced to binary vectors in order to calculate quantitative measures. They are measured by the following: Precision, Recall, F1-score, and Accuracy. Precision is used to determine the percentage of correct pixels that have been spliced, Recall is used to determine the number of correct pixels that have been spliced, F1-score is used to balance Precision and Recall, and Accuracy is used to measure the general accuracy of classification of the entire image.

Overall Pipeline Summary: The entire pipeline combines ELA computation, threshold based segmentation, morphological noise elimination, ground truth mask extraction and metric evaluation. The method offers an effective and interpretable method of detecting the manipulated areas in uncompressed images using compression artifacts and principles of image forensics.

B. Implementation

Algorithm 1: ELA-Based Image Splicing Detection Pipeline

Inputs:
Dataset directories $base\_dir$, $spliced\_dir$, $mask\_dir$
ELA quality $Q = 80$, threshold $T = 25$, kernel size $K = 3$
Number of images $N = 25$
Initialize:
Metrics list $\mathcal{M} \leftarrow \emptyset$
Extract dataset archives if missing
$\mathcal{I} \leftarrow$ all `tif`/`bmp` images in $spliced\_dir$
$count \leftarrow 0$
function GETELA($path, Q$)
    Load image as RGB
    Save image to memory buffer as JPEG (quality Q)
    Reload compressed image
    Compute pixel-wise difference
    Normalize differences to range $[0, 255]$
    return ELA heatmap
end function
function CREATEMASK($mask$)
    Read mask in BGR format
    Extract channels (B, G, R)
    $M \leftarrow (G > R)$
    $M \leftarrow M \wedge (G > B)$
    $M \leftarrow M \wedge (G > 10)$
    Convert M to binary (0/255)
    return M
end function
function PREDICTMASK(E, T, K)
    Convert E to grayscale
    Apply threshold T to obtain P
    if $K > 0$ **then**
        Apply morphological opening on P
        Apply morphological closing on P
    end if
    return P
end function
function HIGHLIGHT(img, P)
    Detect external contours on P
    Draw contours on img
    return highlighted image
end function
procedure RUNPIPELINE
    if $base\_dir$ missing **then**
        Extract dataset ZIP into $base\_dir$
    end if
    if $spliced\_dir$ missing **then**
        Extract `4cam_splc.tar.bz2` archive
    end if
    if $spliced\_dir$ or $mask\_dir$ missing **then**
        return
    end if
    for each image I in $\mathcal{I}$ **do**
        if $count \geq N$ **then**

break
end if
$base \leftarrow$ filename of I without extension
Generate set of candidate mask filenames
$mask\_path \leftarrow$ first existing candidate
if $mask\_path =$ None **then**
continue
end if
Load RGB image X
if load fails **then**
continue
end if
$GT \leftarrow$ CREATEMASK($mask\_path$)
if $GT =$ None **then**
continue
end if
$E \leftarrow$ GETELA(I, Q)
if $E =$ None **then**
continue
end if
$P \leftarrow$ PREDICTMASK(E, T, K)
Highlighted image $\leftarrow$ HIGHLIGHT(X, P)
Flatten GT and P
Compute Precision
Compute Recall
Compute F1-score
Compute Accuracy
Append metrics to $\mathcal{M}$
$count \leftarrow count + 1$
end for
end procedure

III. RESULTS

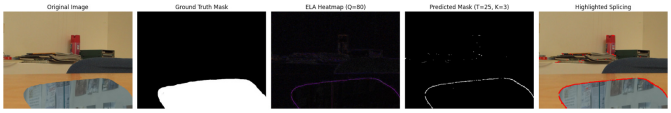

Fig. 1. Example 1: Full ELA-based splicing detection pipeline showing Original Image, Ground Truth Mask, ELA Heatmap ($Q = 80$), Predicted Mask ($T = 25$, $K = 3$), and Highlighted Spliced Region.

Description: This example shows how the proposed system processes a spliced image. The ELA heatmap emphasizes manipulated edges, while the predicted mask identifies the tampered region successfully, with minor noise present.

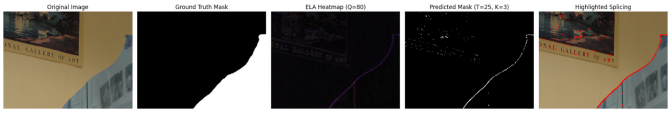

Fig. 2. Example 2: Detection pipeline applied to a second spliced image. The predicted mask captures the major manipulated boundary detected through ELA inconsistencies.

Description: In the second image, ELA highlights the compression variations more strongly, especially along the spliced curve. The prediction mask successfully outlines the manipulated boundary, though some small bright noise patches appear.

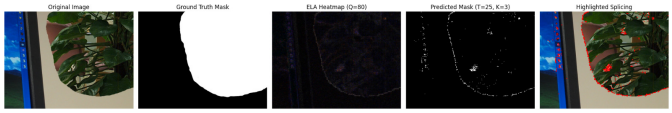

Fig. 3. Example 3: Third demonstration of the detection pipeline. ELA reveals strong edge artifacts around the inserted region, leading to a clear predicted mask.

Description: The third example shows clear ELA edge responses around the manipulated plant region. The predicted mask identifies the spliced area well, although some false positives appear due to textured leaf patterns.

Overall, the examples show that ELA effectively captures manipulation boundaries, though fine-grained detection may vary depending on the complexity of the inserted region.

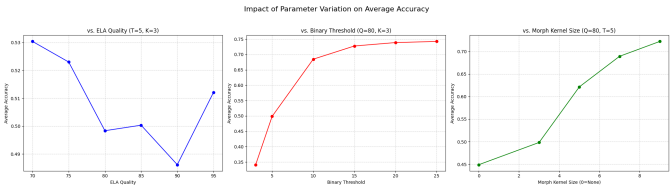

Fig. 4. Impact of parameter variation on average accuracy for ELA quality, binary threshold, and morphological kernel size.

Description: The graphs show that:

- Increasing the **binary threshold** significantly improves accuracy by reducing noise.
- Larger **morphological kernels** smooth the mask and boost performance.
- **ELA quality** fluctuates because too high or too low JPEG qualities distort compression artifacts.

TABLE I
SIMULATION RESULTS FOR PARAMETER VARIATIONS

Parameter Varying	ELA Quality	Binary Threshold	Kernel Size	Avg. Accuracy
ELA_QUALITY	70	5	3	0.530352
ELA_QUALITY	75	5	3	0.522975
ELA_QUALITY	80	5	3	0.498360
ELA_QUALITY	85	5	3	0.500345
ELA_QUALITY	90	5	3	0.486182

Overall, thresholding and morphological refinement have the strongest effect on detection accuracy.

These graphs confirm that thresholding and morphological refinement play the most important role in improving splicing detection accuracy.

IV. CONCLUSION

This work presented an Error Level Analysis (ELA) based approach for detecting spliced regions in uncompressed digital images from the Columbia dataset. The method combines ELA heatmap generation, threshold-based segmentation, and morphological refinement to identify regions with inconsistent

JPEG compression artifacts that typically arise during image manipulation. Experimental results across multiple examples demonstrate that the proposed pipeline is effective in highlighting the manipulated boundaries, with the detected contours aligning closely with the ground truth masks.

The parameter sensitivity analysis revealed that binary threshold and morphological kernel size play a significant role in improving prediction accuracy. Higher thresholds help suppress noise in the ELA heatmap, while larger morphological kernels enhance mask smoothness and reduce fragmentation. In contrast, varying the JPEG resaving quality during ELA computation produced inconsistent performance, indicating that the optimal ELA quality depends on the artifact distribution of each image.

Overall, the proposed method provides an interpretable, lightweight, and computationally efficient solution for splicing detection. While the approach performs well in identifying boundary inconsistencies, certain limitations remain, such as sensitivity to image texture and the presence of false positives in highly detailed regions. Future work may explore integrating deep-learning-based refinement or adaptive thresholding techniques to further enhance robustness. Nonetheless, the results confirm that ELA combined with simple post-processing operations offers a practical and effective foundation for image forgery detection.

REFERENCES

[1] M. F. Jwaid and T. N. Baraskar, "Study and analysis of copy-move & splicing image forgery detection techniques," in *Proc. International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)*, Palladam, India, 2017, pp. 697–702, doi: 10.1109/I-SMAC.2017.8058268.

[2] Z. Tralic, I. Zupancic, S. Grgic, and M. Grgic, "CoMoFoD – New Database for Copy-Move Forgery Detection," in *Proc. 55th International Symposium ELMAR*, Zadar, Croatia, pp. 49–54, Sep. 2013.

[3] H. R. Arpita, S. B. Shwetha, and S. V. Sathyanarayana, "Image Forgery Detection Using Adaptive Oversegmentation and Feature Point Matching," *JNNCE Journal of Engineering & Management (JJEM)*, vol. 3, no. 1, pp. 74–83, Jan.–Jun. 2019.

[4] Columbia Image Splicing Detection Evaluation Dataset, "AuthSplice Uncompressed Image Dataset." Accessed: Nov. 2025. [Online]. Available: https://www.ee.columbia.edu/ln/dvmm/downloads/authsplcuncmp/dlform.html

[5] N. N. Dam, "Splicing Image Detection (GitHub Repository)." Accessed: Nov. 2025. [Online]. Available: https://github.com/NNDam/Splicing-Image-Detection

[6] AICoE, "pyIFD: Python Image Forgery Detection Toolkit (GitHub Repository)." Accessed: Nov. 2025. [Online]. Available: https://github.com/AICoE/pyIFD

[7] N. K. Moudgalya, H. S. Shivashankar, and A. R. Khan, "A Comparative Study of Image Forgery Detection Techniques Using CASIA v2 Dataset," *Engineering, Technology & Applied Science Research*, vol. 13, no. 1, pp. 11345–11351, 2023. [Online]. Available: https://etasr.com/index.php/ETASR/article/view/9593

[8] N. Krawetz, "A Picture's Worth: Digital Image Analysis and Forensics," BlackHat DC Conference, Whitepaper, 2008. [Online]. Available: https://blackhat.com/presentations/bh-dc-08/Krawetz/Whitepaper/bh-dc-08-krawetz-WP.pdf. Accessed: Nov. 2025.