International Journal of Interdisciplinary Innovative Research & Development (IJIIRD)
ISSN: 2456-236X
Vol. 09 Issue 02 | 2025

Image Forgery Detection Using MD5 and OpenCV

Prof. D. J. Manowar[1], Prutha Raut [2], Chetana Kumawat [3], Bhavna Wasankar[4,]
Darshana Khadke[5]

[1]*Prof. D. J. Manowar, Assistant Professor in Dept of CSE, Takshashila Institute of Engineering & Technology, Darapur, Dist Amaravati, Maharashtra, India*

[2,3,4,5] *UG Student, Dept of CSE, Takshashila Institute of Engineering & Technology, Darapur, Dist Amaravati, Maharashtra, India*

DOI: 10.5281/zenodo.15205881

ABSTRACT

With the rise of digital media, image forgery has become increasingly prevalent, posing serious challenges in fields like journalism, forensics, and legal investigations. This project proposes a hybrid method for detecting image forgery by integrating MD5 hashing with advanced OpenCV-based image processing techniques. The approach begins by generating MD5 hash values to quickly verify file integrity; any mismatch between hashes of two images indicates potential tampering at the binary level. For a deeper analysis, the system employs OpenCV to perform grayscale normalization, pixel-level difference mapping, and Structural Similarity Index (SSIM) analysis. These techniques help detect subtle changes in texture, luminance, and structure between the original and the suspect image. Adaptive thresholding and Gaussian blur are applied to enhance heatmaps, highlighting possible forged regions. The system further incorporates HSV color space analysis and frequency domain examination to uncover manipulations such as filtering or contrast adjustments.

Additionally, a custom crop detection algorithm checks for changes in image dimensions to estimate crop percentage. The results are visualized through similarity metrics, forgery probability, and bar graphs. This user-friendly web application provides a reliable, automated solution for authenticating digital images, making it highly suitable for real-world applications.

Keyword: *Image Forgery, MD5, OpenCV, Tampering, Digital Image Forensics, Image Integrity*

1. INTRODUCTION

In today's digital age, ensuring the authenticity of visual content has become increasingly challenging due to the widespread availability of powerful and user-friendly image editing tools. While these tools are valuable in photography, media, and design, they also enable image manipulation that can be misleading or even fraudulent. As a result, there is a growing need for reliable and efficient methods to detect image forgery, especially in areas such as digital forensics, journalism, legal proceedings, and social media.

Image forgery refers to the intentional alteration of an image to misrepresent reality. Common techniques include copy-move tampering, cropping, filtering, and the insertion or removal of elements. Manual detection is often unreliable, especially when alterations are subtle or professionally executed. This has created a demand for intelligent, automated solutions capable of identifying tampered content accurately.

This research presents a web-based image forgery detection system that combines **MD5 hashing** and **OpenCV-based analysis**. MD5 hashing provides a quick and efficient method for verifying file integrity, while OpenCV enables deeper visual comparison using **structural similarity (SSIM)**, **pixel-level analysis**, **heatmaps**, and **crop detection**. The system is designed to be practical, accurate, and user-friendly, offering a robust solution for image verification in real-world scenarios.

2. OBJECTIVES

The main objective of this project is to develop a robust and user-friendly system for detecting image forgery by integrating **MD5 hashing** and **OpenCV-based image analysis techniques**. With the increasing instances of digital image manipulation across social media, journalism, legal investigations, and forensics, the need for reliable tools to verify the authenticity of images has become more critical than ever. This project addresses that need by offering a solution that is both technically effective and easily accessible through a web-based interface.

One key objective is to utilize **MD5 hashing** to perform quick and efficient integrity checks. MD5 generates a unique hash value for each image file; if the hash of a suspect image differs from the original, it indicates tampering at the

binary level. However, as hash-based detection is limited to exact matches, the system also incorporates **OpenCV** for more detailed visual analysis.

The system applies **pixel-level comparison**, **grayscale normalization**, and **Structural Similarity Index (SSIM)** to identify differences between images. It further enhances detection through **heatmap generation**, **adaptive thresholding**, and **Gaussian blurring** to visually highlight manipulated regions. Additionally, the system detects **filter-based edits** using HSV and frequency domain analysis and calculates **cropping percentage** to assess image trimming.

3. METHODOLOGY

The proposed image forgery detection system involves the following major components:

3.1 MD5 Hashing

- MD5 (Message Digest Algorithm 5) is a widely-used cryptographic hashing algorithm that plays a critical role in verifying the integrity of digital images. It transforms an image file into a fixed-length 128-bit hash value, which serves as a unique digital signature for that file.
- The algorithm ensures that every image has a distinct hash value. Even the smallest alteration—such as changing the color of a single pixel, adjusting brightness, or modifying metadata—will result in a completely different hash. This property makes MD5 extremely sensitive and reliable for tamper detection.

Workflow of MD5 Hashing in the System

1. **Original Image Hashing:**
   a. When an original image is uploaded to the system, its MD5 hash is immediately calculated.
   b. This hash value is then stored securely, acting as a reference for any future verification of the image.
2. **Suspect Image Hashing:**
   a. When a potentially tampered or altered image is submitted, the system computes its MD5 hash on the spot.
   b. This ensures the comparison is always based on the most recent version of the image file.
3. **Hash Comparison:**
   a. The system then compares the hash of the suspect image with that of the original image.
   b. If the two hash values match, the image is considered authentic and untampered.
4. **Tampering Detection:**
   a. If the hashes do not match, it strongly indicates that the suspect image has been modified in some way.
   b. This serves as a quick, low-complexity method to flag possible forgery before deeper analysis is performed.

3.2 OpenCV-Based Image Processing

OpenCV is used to analyze the pixel data of the suspect image for signs of tampering. Techniques used include:

- **Pixel-Level Tampering Detection:**
  OpenCV is utilized to examine the pixel data of the suspect image for signs of manipulation. One key technique is edge detection, which helps identify unnatural boundaries created during copy-move or splicing operations. Such edits often produce irregular or inconsistent edges that stand out from the natural flow of the image. Additionally, histogram analysis is used to compare the color distribution across different regions of the image. Inconsistencies in brightness, contrast, or color tones between regions can reveal subtle tampering that may not be visually obvious.
- **Analysis of Texture and Structural Artifacts:**
  To further detect forgery, the system examines blurring and noise patterns, as forged areas often have uneven noise levels or smoothed textures due to editing. These inconsistencies are detectable using OpenCV filters and transformations. Moreover, contour detection is applied to trace the outlines of objects in the image. When content is inserted or removed, the natural shape and alignment of contours may be disrupted. Identifying such irregularities helps highlight manipulated areas, enhancing the system's accuracy in detecting forgeries.

3.3 Implementation

- **Development Environment:** The image forgery detection system is implemented using Python, a powerful and versatile programming language well-suited for image processing and GUI development. Python's extensive libraries such as OpenCV for computer vision and Tkinter or Streamlit for interface design make it ideal for building interactive and functional applications.

- **User Interface (GUI):** The system includes a graphical user interface that allows users to conveniently upload both the original image and the suspect image. This user-friendly interface ensures that individuals without technical expertise can also use the tool with ease**.**
- **Processing and Analysis:** Once the images are uploaded, the application first generates the MD5 hash values for both images and compares them to check for integrity. If a mismatch is found, the system suspects tampering and proceeds to the next step.
- **OpenCV-Based Forgery Detection:** Using OpenCV, the system performs a series of analyses such as edge detection, histogram comparison, noise pattern inspection, and contour detection. These techniques help identify and localize any tampered regions in the suspect image.
- **Output and Notification:** If forgery is detected, the system visually highlights the manipulated areas on the suspect image, such as through bounding boxes or heatmaps. The user is then clearly notified about the forgery, ensuring transparency and reliability.

4.SYSTEM ANALYSIS

The proposed image forgery detection system is a web-based application developed using Python, Flask, and OpenCV libraries. It performs a comprehensive comparison between an original and a suspect image to identify any digital tampering. The analysis is carried out through a combination of image processing techniques, structural analysis, and metadata extraction.

1. Input Handling and Preprocessing:
The system accepts two image inputs—original and suspect—through a web interface. These images are uploaded, stored in a designated directory, and then normalized by converting them to RGB format. This preprocessing ensures consistency for further analysis.

2. Pixel-Level Comparison:
The Structural Similarity Index (SSIM) is used to evaluate the similarity between the two images. The images are first converted to grayscale and resized to match dimensions. SSIM provides a quantitative similarity score, highlighting differences in luminance, contrast, and structure.

3. Heatmap Generation:
Two heatmaps are generated for visual analysis:

- SSIM-Based Heatmap: Created using adaptive thresholding and color mapping to emphasize structural differences.
- Absolute Difference Heatmap: Highlights pixel-level variations using Gaussian blur and threshold masking.

4. Forgery Detection Techniques:
The system identifies various forgery techniques using the following methods:

- Filter Detection: Determines the presence of filters by analyzing the image in both the frequency domain and HSV color space.
- Cropping Detection: Evaluates the difference in dimensions between the two images to estimate the cropping percentage.

5. Metadata and Property Extraction:
The system extracts and displays image metadata including DPI, format, resolution, and file size using the Python Imaging Library (PIL) and EXIF data. These properties support forensic validation of image authenticity.

6. Result Visualization:
The results are presented both numerically and visually. A bar chart is generated to display the similarity score, crop percentage, and forgery estimation. Additionally, heatmaps are displayed to visually indicate tampered regions.

7. Decision Logic:
The final output is based on a composite analysis of the SSIM score, crop ratio, and filter detection. A forgery percentage is calculated to represent the extent of manipulation. If any significant discrepancies are detected, the image is flagged as forged.

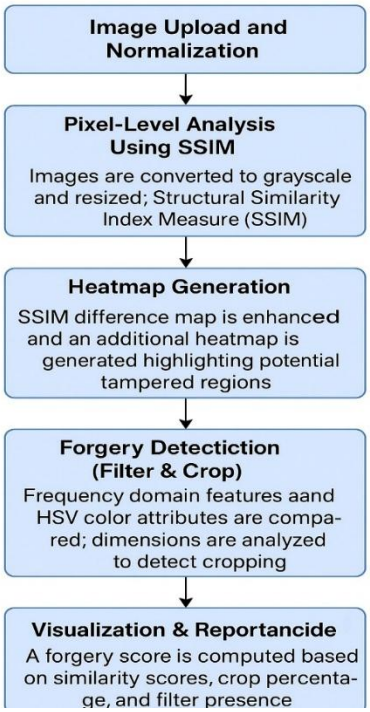

Figure: Image Forgery Detection Workflow Using SSIM and Feature Analysis

5. RESULTS AND DISCUSSION

The proposed system successfully detects image forgery using a multi-layered approach incorporating Structural Similarity Index (SSIM), pixel-level difference analysis, and image property comparison. When original and suspect images are uploaded, the system normalizes both, compares their structural integrity using SSIM, and generates a detailed heatmap to highlight altered regions. Forgery is further examined through detection of filters and cropping based on frequency domain and color analysis, as well as dimensional changes.

Experimental results reveal that even subtle modifications, such as brightness adjustments or partial cropping, are effectively detected. The SSIM similarity score inversely correlates with forgery percentage, providing quantitative insight into tampering. Enhanced heatmaps and difference maps visually represent the altered areas, offering clarity for analysis. Additionally, the system outputs metadata like DPI, image format, and size, contributing to forensic-level scrutiny.

6. CONCLUSIONS

The proposed image forgery detection system successfully identifies and highlights tampered regions using a combination of SSIM-based analysis, pixel-level difference heatmaps, and image metadata comparison. It effectively detects common forgery techniques such as cropping, filtering, and structural alterations. By providing both visual evidence and quantitative metrics, the system ensures accurate and reliable results. This comprehensive approach makes it a practical and efficient tool for digital image authentication in forensic, academic, and security-related applications.

7. REFERENCES

[1] Areeb Hassan Mukhdoomi; Umad Bashir Sofi; Mr. K. Suresh: "Python Image Forgery Detection Using MD5 and OpenCV," SSRN, 2023. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4852380

[2] P. Appalanaidu, S. P. Sanjana, and J. S. Jyothika, "Image Forgery Detection Using OpenCV and MD5," Journal of Operating Systems Development & Trends, vol. 1998.: http://www.journal-iiieindia.com/1_apr_23/30_online.pdf

[3] Prof. Chandrashekhar Badgujar, Ms. Roshani Nakti, Ms. Riddhi Shinde,: IMAGE FORGERY DETECTION USING OPEN-CV AND MD5 ALGORITHM e-ISSN: 2582-5208 International Research Journal of

Modernization in Engineering Technology and Science (Peer-Reviewed, Open Access, Fully Refereed International Journal) Volume:06/Issue:04/April-2024 Impact Factor- 7.868 www.irjmets.com

[4] M.Lohithdakshan, Maithili Jha, S. Maitri: "Image forgery detection using OpenCV and MD5," JETIR, vol. 2312562, 2023. : https://www.jetir.org/papers/JETIR2312562.pdf

[5] Xiao, B.; Wei, Y.; Bi, X.; Li, W.; Ma, J. Image splicing forgery detection combining coarse to a refined convolutional neural network and adaptive clustering. Inf. Sci. 2020, 511, 172–191.

[6] Kwon, M.J.; Yu, I.J.; Nam, S.H.; Lee, H.K. CAT-Net: Compression Artifact Tracing Network for Detection and Localization of Image Splicing. In Proceedings of the 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 5–9 January 2021; pp. 375–384.

[7] Zheng, L.; Zhang, Y.; Thing, V.L. A survey on image tampering and its detection in real-world photos. J. Vis. Commun. Image Represent. 2019, 58, 380–399.Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 5–9 January 2021; pp. 375–384.

[8] R. Shao and E. J. Delp, "Forensic Scanner Identification Using Machine Learning," 2020 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI), Albuquerque, NM, USA, 2020, pp. 1-4, doi: 10.1109/SSIAI49293.2020.9094618.

[9] Y. Q. Shi, C. Chen, and W. Chen, "A natural image model approach to splicing detection," Proceedings of the 9th workshop on Multimedia & Security, pp. 51–62, September 2007, Dallas, TX.

[10] Sevinc Bayram, Husrev Taha Sencar, and Nasir Memon, "An efficient and robust method for detecting copy-move forgery," Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp.1053–1056, April 2009, Taipei, Taiwan.

[11] Miss. Devika M. Shelke, Miss. Roshani S. Bhojane, Miss. Tina B. Madane, Mr. Pratik N. Gawande, Prof. D.J. Manowar, Prof. S.S. Dubey, '' Data Store and Multi-Keyword Search on Encrypted Cloud Data'', ISSN 2320–088X IJCSMC, Vol. 3, Issue. 4, April 2014, pg.1227 – 1232 www.ijcsmc.com .