STM JOURNALS

Journal of
Operating Systems
Development & Trends

ISSN: 2454-9355
Volume 11, Issue 1, 2024
January–April
DOI (Journal): 10.37591/JoOSDT

https://journals.stmjournals.com/joosdt

Review JoOSDT

# Detecting Fake Images with Python: A Simple Approach Using OpenCV and MD5

Nivedha R.[1,*], Ranjana S.[2]

***Abstract***

*In today's modern era, ensuring the authenticity of pictures is crucial. This article presents a straightforward method for identifying fake pictures using the widely used OpenCV and MD5 technologies. OpenCV helps us examine pictures for irregularities such as abnormal colors or shapes, while MD5 is a unique digital fingerprint for each image. Our approach combines these tools to create a robust fraud detection system. OpenCV carefully examines various aspects of pictures, flagging potential areas of manipulation. In addition, MD5 generates a unique code or hash that effectively represents the entire image. We tested our method on various datasets containing both genuine and fake pictures. The results demonstrate that our system accurately detects manipulated regions and verifies authenticity by comparing unique MD5 hashes. This research provides a practical solution for ensuring image integrity in law, medicine, and interactive media.*

**Keywords:** Digital images, image forgery, authenticity, conventional techniques, OpenCV, detecting fraud, and MD5

## INTRODUCTION

In the digital age we live in today, the widespread availability and affordability of digital cameras has resulted in a significant increase in photo-taking and sharing. However, the ease of access to image editing tools has led to the misuse of altering photos, causing the spread of misinformation and the creation of fraudulent images. This is a serious issue as manipulated visuals can have irreversible consequences. The purpose of this study is to address this problem by reducing the occurrence of fraudulent images and improving the ability to differentiate between authentic and fake photos for more accurate identification.

*The study discusses two common types of image forgeries: Image Splicing*, which involves combining sections from different photos to create a misleading composite image, and *copy-move*, which entails copying and pasting a portion of an image within the same image to either hide or duplicate content. Detecting these types of forgeries requires advanced techniques, such as analyzing variations in texture, color, and lighting for image splicing, and utilizing digital image processing methods for copy-move detection [1].

***Author for Correspondence**
Nivedha R.
E-mail: nivedhadinesh3@gmail.com

[1]Student, Department of Computer Science, Anna Adarsh College for Women, Chennai, India
[2]Assistant Professor, Department of Computer Science, Anna Adarsh College for Women, Chennai, India

Received Date: April 09, 2024
Accepted Date: April 12, 2024
Published Date: May 03, 2024

**Citation:** Nivedha R., Ranjana S. Detecting Fake Images with Python: A Simple Approach Using OpenCV and MD5. Journal of Operating Systems Development & Trends. 2024; 11(1): 45–49p.

### Problem Definition

The issue addressed in *"Detecting Fake Images with Python:* A Simple Approach Using OpenCV and MD5" arises from the widespread use of image editing tools, which has created an urgent need for a straightforward yet dependable method to differentiate between authentic and manipulated images. With the easy availability of editing tools,

 

there is an increasing risk of deceptive visual content. The challenge lies in developing a user-friendly approach, ensuring that even individuals with limited technical expertise can confidently identify fake images.

This problem revolves around the prevalence of manipulated images, the simplicity of manipulation tools, and the necessity for a reliable and uncomplicated detection method. The objective is to incorporate OpenCV for image processing and MD5 hashing for integrity verification, enabling the detection of fake images to be practical and efficient in real-world scenarios. The research study not only aims to advance theoretical concepts but also offers a user-friendly solution for practical implementation, emphasizing simplicity and effectiveness in detecting image forgery.

**Objectives**

The objective of this project is to develop a reliable system for detecting image forgery by integrating MD5 hashing and OpenCV. The main goals include verifying the authenticity of images by comparing their MD5 hash values and effectively identifying any instances of image forgery.

## PROJECT DETAILS

### Image Retrieval

Using the OpenCV computer vision library, the system retrieves both the original image (Figure 1) and the image being examined (Figure 2) [2].

### Image Encoding

The two images are encoded using the Message Digest5(MD5) Hashlib module. MD5 is a commonly adopted cryptographic hash function that generates unique hash values for each input, ensuring a strong representation.

### Comparison

The project carefully compares the MD5 hash values of Figure 1 and Figure 2. A matching hash value indicates that the images are identical, indicating no forgery. On the other hand, different hash values indicate potential alterations or forgery.

**Figure 1.** Original.

**Figure 2.** Forged.

 

**Result Display**

The results of the forgery detection process are presented on the admin page, providing a user-friendly display that indicates whether the tested image has undergone any form of forgery. This transparent presentation enhances the ease of interpretation for users.

By fulfilling these objectives, the project establishes an efficient and user-friendly image forgery detection system. The seamless integration of MD5 hashing and OpenCV capabilities empowers users to identify potential alterations swiftly and accurately in images, contributing to enhanced trust and reliability in digital image verification.

## METHODOLOGY

### Image Preprocessing

At the initial stage of our approach, we utilize the *OpenCV library* to carry out meticulous image preprocessing tasks. These tasks encompass noise reduction, resizing, and color normalization. The purpose of these preprocessing steps is to optimize the subsequent forgery detection algorithms. By refining the input data, we ensure that it is conducive to accurate analysis.

### Forgery Detection

Our forgery detection algorithm seamlessly integrates into the OpenCV framework. It employs sophisticated techniques such as edge detection, texture analysis, and feature extraction. The objective is to systematically identify regions within an image that may have undergone manipulation. By discerning intricate patterns and anomalies, the algorithm effectively distinguishes between authentic and forged components. This contributes to a nuanced understanding of potential alterations.

### MD5 Hashing

Maintaining the integrity of the identified regions is of utmost importance in our methodology. To achieve this, we introduce the robust MD5 hashing technique. This cryptographic process generates unique hashes for both the original and processed images. By scrutinizing any disparities between these hashes, our methodology provides a reliable indicator of potential tampering or alterations. This step adds an extra layer of security and verifiability to the forgery detection process [3].

In summary, our comprehensive methodology integrates advanced image preprocessing with a sophisticated forgery detection algorithm within the OpenCV framework. The incorporation of MD5 hashing ensures not only the accuracy of the detection process but also guarantees the integrity of identified regions. This makes our approach a robust solution for image forgery detection. Extensive experimentation on diverse datasets has validated the efficacy of this methodology, demonstrating its potential for addressing the challenges posed by image manipulation in various applications.

## IMPLEMENTATION

Python scripts use OpenCV to resize, crop, and adjust colors in image processing. The MD5 algorithm is utilized to create hash values for comparing the original and processed images, allowing for an uncomplicated yet efficient method of detecting images [4].

## EXPERIMENTAL RESULTS

During the experimental phase, we carefully evaluated our method using a wide range of images that were known to be manipulated. The results showed that our method was highly accurate in detecting common manipulations such as scaling, rotation, and changes in color composition. Additionally, our approach proved to be very effective in detecting subtle changes within images, demonstrating its ability to identify different types of manipulation. These findings confirm that our method has great potential for reliable image forensics, which is a significant step toward ensuring the credibility of digital visual content [5].

 

## FOCAL POINTS OF DETECTING FAKE IMAGES WITH PYTHON: A SIMPLE APPROACH USING OPENCV AND MD5

### Robust Detection Algorithm

By incorporating OpenCV, a sophisticated forgery detection algorithm can be implemented. This algorithm utilizes techniques like edge detection, texture analysis, and feature extraction to accurately identify forged images [6].

### Versatility of OpenCV

OpenCV is a versatile computer vision library that offers a wide range of tools and functions. This makes it suitable for various image processing tasks beyond forgery detection, expanding its utility.

### MD5 Hashing for Data Integrity

The utilization of MD5 hashing ensures the integrity of the detected regions. It generates unique hash values for both the original and processed images, enhancing the reliability of the forgery detection process through cryptographic techniques [7].

### Ease of Implementation

Python is renowned for its simplicity and readability. Implementing image forgery detection using Python, OpenCV, and MD5 is relatively straightforward. This accessibility makes it available to a broader audience, including researchers and practitioners.

### Performance Across JPEG and Non-JPEG Formats

The proposed method demonstrates excellent performance in both JPEG and non-JPEG image localization. This capability allows for handling different image formats commonly encountered in real-world scenarios [8].

## IMPEDIMENTS OF DETECTING FAKE IMAGES WITH PYTHON: A SIMPLE APPROACH USING OPENCV AND MD5

### Limited to Inactive Images

The current approach primarily focuses on detecting fraud in inactive images. It may not be suitable for identifying fraudulent activities in dynamic content such as videos or live streams.

### Sensitivity to Preprocessing Quality

The effectiveness of fraud detection can be affected by the quality of image preprocessing. Inconsistent or inadequate preprocessing techniques may impact the accuracy of the algorithm.

### Potential False Positives

Although MD5 hashing is reliable for integrity verification, it can generate false positives in fraud detection if the original image undergoes legitimate modifications that are not malicious [9].

### Dependence on OpenCV Updates

The success of this strategy relies on the features and updates provided by the OpenCV library. Changes in the library's functionality or updates may impact the performance and accuracy of the fraud detection algorithm.

### Resource Intensive

Image processing tasks, especially those involving complex algorithms, can be computationally demanding. Large datasets or high-resolution images may require significant computational resources [10].

## CONCLUSION

In this study, a straightforward yet impactful method for detecting counterfeit images is introduced. By employing both OpenCV and the MD5 algorithm, this approach proves to be highly capable of

 

identifying different types of image alterations. Moreover, it is a lightweight and computationally efficient solution that seamlessly integrates into preexisting image verification systems. Future studies might consider expanding this method to tackle more complex manipulations and incorporating machine learning techniques to enhance accuracy.

## REFERENCES

1. Appalanaidu P, Sanjana P, Jyothika S, Students T. Image forgery detection using OpenCV and MD5. Ind Eng J. 2023 Apr;52(4):221-6. DOI: 10.36893.IEJ.2023.V52I04.221-226.
2. Javatpoint.com. (2021). Image Forgery Detection Using Machine Learning. [online] Javatpoint. Available from: https://www.javatpoint.com/image-forgery-detection-using-machine-learning.
3. Zanardelli M, Guerrini F, Leonardi R, Adami N. Image forgery detection: a survey of recent deep-learning approaches. Multimedia Tools and Applications. 2023 May;82(12):17521-66.
4. Zheng L, Zhang Y, Thing VLL. A survey on image tampering and its detection in real-world photos. J Vis Commun Image Represent. 2019;58:380-99. DOI: 10.1016/j.jvcir.2018.12.022.
5. Bayram S, Taha Sencar HT, Memon N. An efficient and robust method for detecting copy-move forgery. 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, Taiwan, 2009, pp. 1053-1056. DOI: 10.1109/ICASSP.2009.4959768.
6. Divya G. (2023). Python image forgery detection using OpenCV. Ameerpet, Vol. MD5. (2023). Available from: https://www.ameerpet.org/python-image-forgery-detection-using-md5-opencv.html.
7. Kristianto GY, Topic G, Aizawa A. MCAT math retrieval system for NTCIR-12 MATHIR task. Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies, June 7-10, Tokyo, Japan. 2016, pp. 323-330.
8. Laroudie C, Bursuc A, Ha ML, Franchi G. Improving CLIP Robustness with Knowledge Distillation and Self-Training. arXiv Preprint ArXiv:2309.10361. 2023 Sep 19.
9. Sreekumar KS. (2023). Exploring the purpose of a notary: Ensuring trust and authenticity-Holborn notary. Holborn notary. Available from: https://holbornnotary.com/exploring-the-purpose-of-a-notary-ensuring-trust-andauthenticity/.
10. FasterCapital. (2024). Choosing best AI detector key features and recommendations. FasterCapital. Available from: https://fastercapital.com/content/Choosing-best-ai-detector-key-features-and-recommendations.html.

 