JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

IMAGE TAMPERING DETECTION USING DEMOSAICING ARTIFACTS

J Dhivagar[1], Pavithra A [2]

[1]J Dhivagar M.Sc, Department of Computer Science and Engineering, Dr. MGR Educational and Research Institute, Chennai, India
[2]Pavithra A Faculty, Centre of Excellence in Digital Forensics, Dr. MGR Educational and Research Institute, Chennai, India

Abstract:

Even novice people can nowadays do convincing forged images.Detecting picture manipulation is critical to confirming the validity of digital photographs, particularly in areas like law, journalism, and security where maintaining image integrity is vital. An efficient way to identify tampering is to examine demosaicing artifacts. Using a color filter array (CFA), such as the Bayer pattern, digital cameras use demosaicing to rebuild full-color images from the partial color samples that are output by image sensors. Characteristic artifacts are left in the image by this method.This work uses these artifacts of demosaicing to locate areas of digital photographs that have been tampered with. The natural demosaicing patterns are frequently broken or changed when an image is altered. We are able to identify discrepancies suggestive of manipulation by analyzing the consistency and statistical characteristics of these artifacts.

Keywords
Detecting picture,Image integrity,Demosaicing artifacts,Color Filter Array (CFA),Bayer pattern,Image sensors.

I. INTRODUCTION

The digital revolution has incontrovertibly converted our world, and the realm of visual communication is no exception. The ease with which images can be captured, manipulated, and circulated online has fostered a growing concern – the proliferation of tampered images. These altered illustrations can have far- reaching consequences, impacting everything from news credibility to legal proceedings. Accordingly, the development of robust and dependable image tampering discovery ways has come an critical precedence. One promising approach in this field leverages the essential characteristics of digital cameras.

utmost ultramodern cameras prisoner color information using a Color Filter Array(CFA) deposited over the image detector. Each pixel on the CFA captures only one color(red, green, or blue). To produce the full- color image we see, a process called demosaicing interpolates the missing color values for each pixel. Different demosaicing algorithms introduce specific vestiges into the final image, which can be subtle but sensible(1).

This veritably characteristic of demosaicing – the preface of vestiges – has sparked a new direction in image tampering discovery. Experimenters have observed that tampering operations, like splicing or object cloning, can disrupt the natural patterns of these vestiges. By assaying inconsistencies in demosaicing vestiges across different image regions, it becomes possible to identify implicit manipulation(2, 3). This literature review delves into this instigative area, exploring the theoretical foundations, being ways, and promising exploration trends in image tampering discovery using demosaicing vestiges.

This preface provides environment, highlights the significance of image tampering discovery, and introduces the conception of demosaicing vestiges and their implicit for this purpose. It also acknowledges applicable sources for farther disquisition.

REVIEW OF LITERATURE

L. Wang and G. Jeon, et al, Nov 2015 [7]demonstrates Bayer Pattern CFA Demosaicking Based on Multi-Directional Weighted Interpolation and Guided Filter,. Abstract: In this letter, we proposed a new framework for color image demosaicking by using different strategies on green (G) and red/blue (R/B) components. Firstly, for G component, the missing samples are estimated by eight-direction weighted interpolation via exploiting spatial and spectral correlations of neighboring pixels. The G plane can be well reconstructed by considering the joint contribution of pre-estimations along eight interpolation directions with different weighting factors. Secondly, we estimate R/B components using guided filter with the reconstructed G plane as guidance image. Simulation results verify that, the proposed framework performs better than state-of-the-art demosaicking methods in term of color peak signal-to-noise ratio (CPSNR) and feature similarity index measure (FSIM), as well as higher visual quality.

L. Bondi, S. Lameri, D. Güera, P. Bestagini, E. J. Delp and S. Tubaro, et al, 2015 [8] explains Tampering Detection and Localization Through Clustering of Due to the rapid proliferation of image capturing devices and user-friendly editing software suites, image manipulation is at everyone's hand. For this reason, the forensic community has developed a series of techniques to determine image authenticity. In this paper, we propose an algorithm for image tampering detection and localization, leveraging characteristic footprints left on images by different camera models. The rationale behind our algorithm is that all pixels of pristine images should be detected as being shot with a single device. Conversely, if a picture is obtained through image composition, traces of multiple devices can be detected. The proposed algorithm exploits a convolutional neural network (CNN) to extract characteristic camera model features from image patches. These features are then analyzed by means of iterative clustering techniques in order to detect whether an image has been forged, and localize the alien region.

G. Cao, Y. Zhao, R. Ni, L. Yu and H. Tian, et al, [9] demonstrates In digital image forensics, prior works are prone to the detection of malicious tampering. However, there is also a need for developing techniques to identify general content-preserved manipulations, which are employed to conceal tampering trails frequently. In this paper, we propose a blind forensic algorithm to detect median filtering (MF), which is applied extensively for signal denoising and digital image enhancement. The probability of zero values on the first order difference map in texture regions can serve as MF statistical fingerprint, which distinguishes MF from other operations. Since anti-forensic techniques enjoy utilizing MF to attack the linearity assumption of existing forensics algorithms, blind detection of the non-linear MF becomes especially significant. Both theoretically reasoning and experimental results verify the effectiveness of our proposed MF forensics scheme.

H. Farid, et al,march 2009 [10] explains When creating a digital forgery, it is often necessary to combine several images, for example, when compositing one person's head onto another person's body. If these images were originally of different JPEG compression quality, then the digital composite may contain a trace of the original compression qualities. To this end, we describe a technique to detect whether the part of an image was initially compressed at a lower quality than the rest of the image. This approach is applicable to images of high and low quality as well as resolution.

S. Lameri, D. Güera, P. Bestagini, E. J. Delp and S. Tubaro, et al, 2015 [11] explains Tampering Detection and Localization Through Clustering of Due to the rapid proliferation of image capturing devices and user-friendly editing software suites, image manipulation is at everyone's hand. For this reason, the forensic community has developed a series of techniques to determine image authenticity. In this paper, we propose an algorithm for image tampering detection and localization, leveraging characteristic footprints left on images by different camera models. The rationale behind our algorithm is that all pixels of pristine images should be detected as being shot with a single device. Conversely, if a picture is obtained through image composition, traces of multiple devices can be detected. The proposed algorithm exploits a convolutional neural network (CNN) to extract characteristic camera model features from image patches. These features are then analyzed by means of iterative clustering techniques in order to detect whether an image has been forged, and localize the alien region.

Popescu, Alin C. and Farid, Hany, et al, [12] describe an efficient technique that automatically detects duplicated regions in a digital image. This technique works by first applying a principal component analysis to small fixed-size image blocks to yield a reduced dimension representation. This representation is robust to minor variations in the image due to additive noise or lossy compression. Duplicated regions are then detected by lexicographically sorting all of the image blocks. We show the efficacy of this technique on credible forgeries, and quantify its robustness and sensitivity to additive noise and lossy JPEG compression.

S. Ye, Q. Sun and E. -C. Chang, et al, 2007 [13] Digital images can be forged easily with today's widely available image processing software. In this paper, we describe a passive approach to detect digital forgeries by checking inconsistencies of blocking artifact. Given a digital image, we find that the blocking artifacts introduced during JPEG compression could be used as a "natural authentication code". A blocking artifact measure is then proposed based on the estimated quantization table using the power spectrum of the DCT coefficient histogram. Experimental results also demonstrate the validity of the proposed approach.

II. RESEARCH METHODOLOGY

In this project, RGB algorithm is used for the detection of the tampered images. Firstly, RGB algorithm stands for R(red), G(green), B(blue). The RGB color model is one of the most widely used color representation method in computer graphics. It use a color coordinate system with three primary colors. The RGB color model is an additive color model in which the red, green and blue primary colors of light are added together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the three additive primary colors, red, green, and blue. The main purpose of the RGB color model is for the sensing, representation, and display of images in electronic systems, such as televisions and computers, though it has also been used in conventional photography and colored lighting. Before the electronic age, the RGB color model already had a solid theory behind it, based in human perception of colors. RGB is a device-dependent color model: different devices detect or reproduce a given RGB value differently, since the color elements (such as phosphors or dyes) and their response to the individual red, green, and blue levels vary from manufacturer to manufacturer, or even in the same device over time.

Thus an RGB value does not define the same color across devices without some kind of color management. Typical RGB input devices are color TV and video cameras, image scanners, and digital cameras. Typical RGB output devices are TV sets of various technologies (CRT, LCD, plasma, OLED, quantum dots, etc.), computer and mobile phone displays, video projectors, multicolor LED displays and large screens such as the Jumbotron. Color printers, on the other hand, are not RGB devices, but subtractive color devices typically using the CMYK color model. As shown in fig1

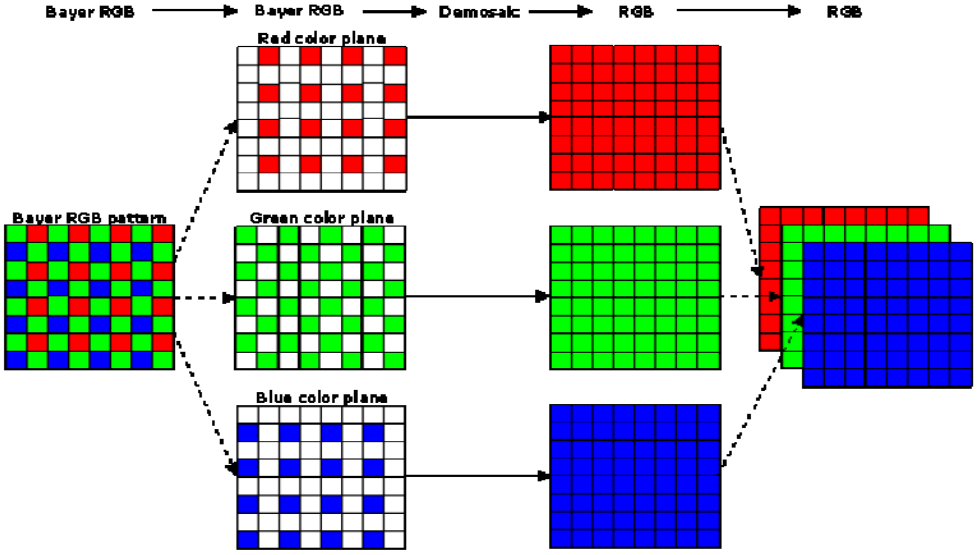


fig 1.Bayer RGB pattern

3.1 Data Collection and Understanding

The suggested image has been now analysed and considered to analyse the image for knowing their RGB factors(Red, Green, Blue) and the HSV i.e, the Hue Saturation Value. Now then the given image has been extracted using the RGB factors and the hue channels to calculate the median value of the image.

3.2Algorithm Implementation

The suggested image has been now set to calculate the hue median value by knowing the HSV value and the algorithm used here is the numpy as np and CV2 which runs on the python and here were implanting the algorithm and the image has been added to the trained dataset.

3.3 Demosaicing Algorithm

The HSV (Hue, Saturation, Value) color space is chosen because it separates color information (hue) from intensity information (value), making it easier to analyze colors without the influence of lighting and shadows. The conversion from BGR (Blue, Green, Red) to HSV is done using OpenCV's cv2.cvtColor function. Using the HSV color space allows for more accurate detection of color anomalies, which is crucial for identifying tampered regions that may have different color characteristics compared to the rest of the image.

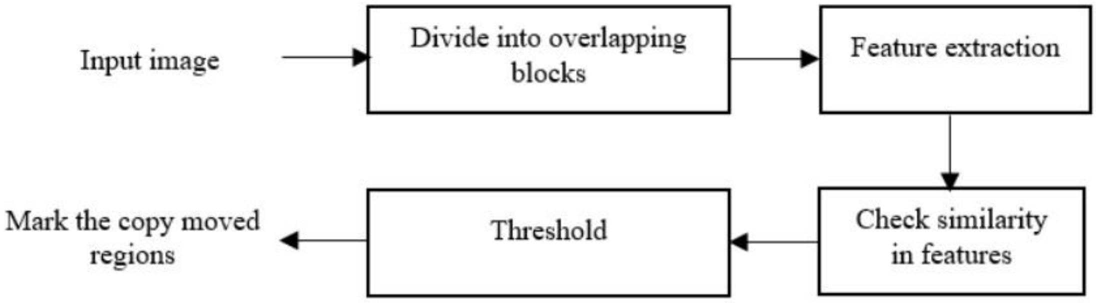
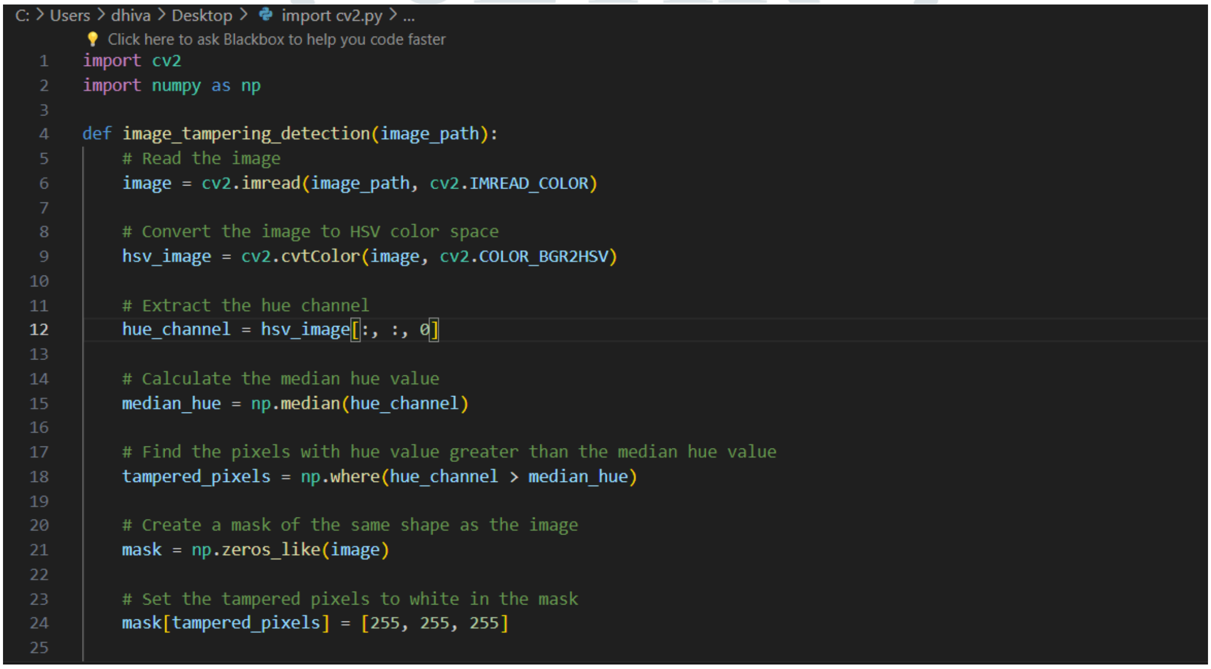

Input image
Divide into overlapping blocks
Feature extraction
Check similarity in features
Threshold
Mark the copy moved regions

fig 2. Image explaining the workflow

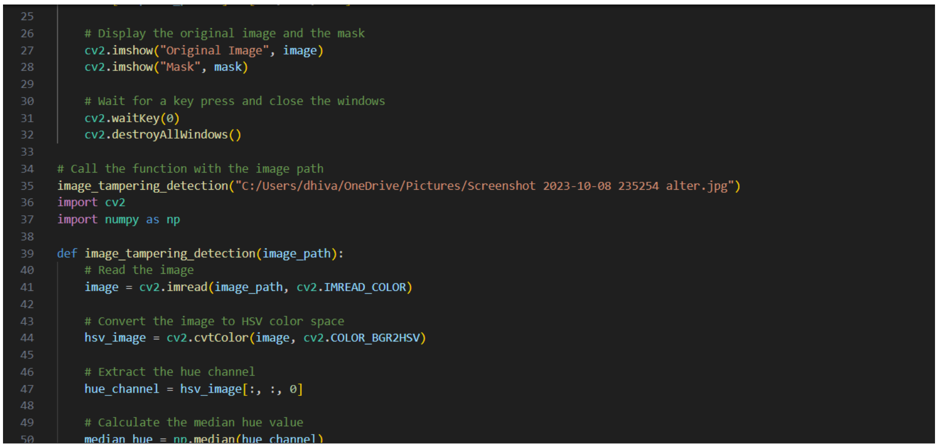
In the below shown figure we will give the code to read and analyse the image and by using CV2 the image will be read and then the image will processed to detect and separate the HSV value from the image i.e. the hue seperation value of thew image then the values of the image's hue value will be detected


C: > Users > dhiva > Desktop > import cv2.py > ...
Click here to ask Blackbox to help you code faster

```
import cv2
import numpy as np

def image_tampering_detection(image_path):
    # Read the image
    image = cv2.imread(image_path, cv2.IMREAD_COLOR)

    # Convert the image to HSV color space
    hsv_image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)

    # Extract the hue channel
    hue_channel = hsv_image[:, :, 0]

    # Calculate the median hue value
    median_hue = np.median(hue_channel)

    # Find the pixels with hue value greater than the median hue value
    tampered_pixels = np.where(hue_channel > median_hue)

    # Create a mask of the same shape as the image
    mask = np.zeros_like(image)

    # Set the tampered pixels to white in the mask
    mask[tampered_pixels] = [255, 255, 255]

```

fig 3 displaying the source code

The below shown figure shows where the code uses the original image that user has inputted and it has been calling the function from the images path and here the CV2 will start to analyse the image for its colour values and hue values following by the HSV color space and extracting the HUE channel and calculating the median value.



```
    # Display the original image and the mask
    cv2.imshow("Original Image", image)
    cv2.imshow("Mask", mask)

    # Wait for a key press and close the windows
    cv2.waitKey(0)
    cv2.destroyAllWindows()

# Call the function with the image path
image_tampering_detection("C:/Users/dhiva/OneDrive/Pictures/Screenshot 2023-10-08 235254 alter.jpg")
import cv2
import numpy as np

def image_tampering_detection(image_path):
    # Read the image
    image = cv2.imread(image_path, cv2.IMREAD_COLOR)

    # Convert the image to HSV color space
    hsv_image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)

    # Extract the hue channel
    hue_channel = hsv_image[:, :, 0]

    # Calculate the median hue value
    median_hue = np.median(hue_channel)
```

fig 4 displaying the source code

The below shown figure shows the image which has been entered with detailed information of the image’s path ,where the image have been actually stored. Here the iamge has been altered and tampered which cant be seen through the normal vision of the human naked eye.

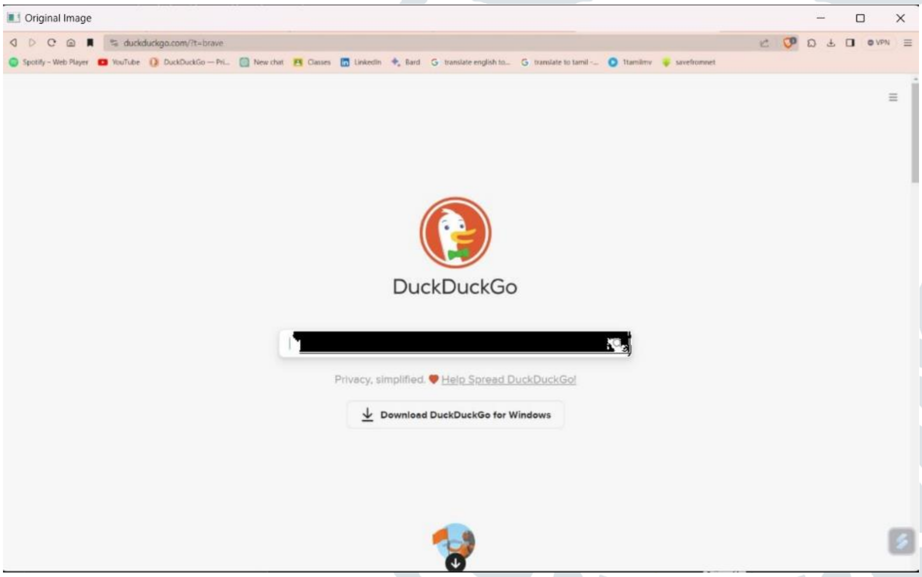

Original Image
duckduckgo.com/?t=brave
DuckDuckGo
Privacy, simplified. Help Spread DuckDuckGo!
Download DuckDuckGo for Windows

fig 5 Tampered Image

In the below figure it is showing that the altered or the tampered areas in this image have been displayed as the output and comparing to the real image that has been used . The tampered part in the iamge are differentiated in white space and the non tampered region have been in black space .

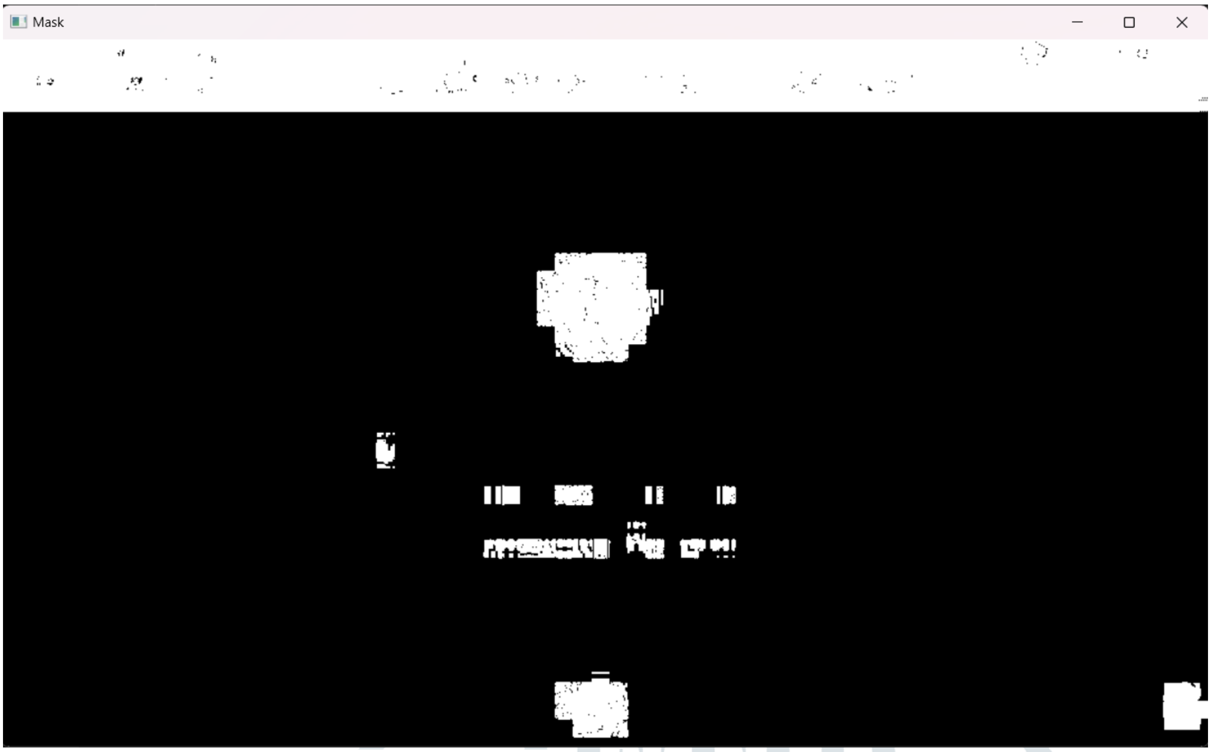


fig 6 displaying the result of the tampered areas in white

CONCLUSION

Image tampering detection using demosaicing artifacts is a powerful approach that leverages the inherent properties of digital image formation. By analysing the consistency of demosaicing artifacts, forensic experts can detect and localize tampered regions, enhancing the reliability and authenticity of digital images. As image manipulation techniques evolve, continued research and development of robust detection methods will be essential to stay ahead of sophisticated tampering attempts.
Another method involves the use of machine learning algorithms trained on the specific demosaicing artifacts of different camera models. By comparing the artifacts in the suspect image against a database of known patterns, these algorithms can identify discrepancies that suggest manipulation. Despite its effectiveness, this approach has limitations. Advanced tampering techniques may mimic the demosaicing process to cover up inconsistencies. Additionally, images that undergo multiple compression stages or are captured by devices with non-standard demosaicing algorithms can pose challenges for detection. 27

In conclusion, leveraging demosaicing artifacts for image tampering detection offers a robust tool in digital forensics. By focusing on the intrinsic properties introduced during the image formation process, investigators can uncover hidden manipulations. Ongoing advancements in analytical techniques and machine learning promise to enhance the accuracy and reliability of this method, making it an indispensable asset in the fight against digital fraud and misinformation.

REFERENCES

[1]. H. Farid, "Image forgery detection," in IEEE Signal Processing Magazine, vol. 26, no. 2, pp. 16-25, March 2009, doi: 10.1109/MSP.2008.931079.

[2]. Zhen, R., Stevenson, R. (2015). Image Demosaicing. In: Celebi, E., Lecca, M., Smolka, B. (eds) Color Image and Video Enhancement. Springer, Cham.

[3]. Xin Li, Bahadir Gunturk, Lei Zhang, "Image demosaicing: a systematic survey," Proc. SPIE 6822, Visual Communications and Image Processing 2008, 68221J

[4]. E. A. Armas Vega, E. González Fernández, A. L. Sandoval Orozco and L. J. García Villalba, "Passive Image Forgery Detection Based on the Demosaicing Algorithm and JPEG Compression," in IEEE Access, vol. 8, pp. 11815-11823, 2020,

[5]. A. C. Gallagher and Tsuhan Chen, "Image authentication by detecting traces of demosaicing," 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Anchorage, AK, USA, 2008, pp

[6]. P. Ferrara, T. Bianchi, A. De Rosa and A. Piva, "Image Forgery Localization via Fine-Grained Analysis of CFA Artifacts," in IEEE Transactions on Information Forensics and Security, vol. 7, no. 5, pp. 1566-1577, Oct. 2012, doi: 10.1109/TIFS.2012.2202227

[7]. L. Wang and G. Jeon, "Bayer Pattern CFA Demosaicking Based on Multi-Directional Weighted Interpolation and Guided Filter," in IEEE Signal Processing Letters, vol. 22, no. 11, pp. 2083-2087, Nov. 2015, doi: 10.1109/LSP.2015.2458934

[8]. L. Bondi, S. Lameri, D. Güera, P. Bestagini, E. J. Delp and S. Tubaro, "Tampering Detection and Localization Through Clustering of Camera-Based CNN Features," 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 2017

[9]. G. Cao, Y. Zhao, R. Ni, L. Yu and H. Tian, "Forensic detection of median filtering in digital images," 2010 IEEE International Conference on Multimedia and Expo, Singapore, 2010, pp. 89-94, doi: 10.1109/ICME.2010.5583869.

[10]. H. Farid, "Exposing Digital Forgeries From JPEG Ghosts," in IEEE Transactions on Information Forensics and Security, vol. 4, no. 1, pp. 154-160, March 2009, doi: 10.1109/TIFS.2008.2012215.

[11]. L. Bondi, S. Lameri, D. Güera, P. Bestagini, E. J. Delp and S. Tubaro, "Tampering Detection and Localization Through Clustering of Camera-Based CNN Features," 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 2017

[12]. Popescu, Alin C. and Farid, Hany, "Exposing Digital Forgeries by Detecting Duplicated Image Regions" (2004). Computer Science Technical Report TR2004-515.

[13]. S. Ye, Q. Sun and E. -C. Chang, "Detecting Digital Image Forgeries by Measuring Inconsistencies of Blocking Artifact," 2007 IEEE International Conference on Multimedia and Expo, Beijing, China, 2007, pp. 12-15, doi: 10.1109/ICME.2007.4284574.