Webology (ISSN: 1735-188X)
Volume 19, Number 2, 2022

Efficient Analysis Of Manipulated Image Processing

Gavendra Singh[1] , Faizur Rashid[2] , Afendi Abdi[3]

Department of Software Engineering[1,3]

Department of Computer Science[2]

College of computing and Informatics Haramaya University 138, Dire Dawa. Ethiopia.

Abstract

The fast progress in engineered image generation and manipulation has now gone to a point where it raises huge worries on the suggestion on the public. Best-case scenario, this prompt lost trust in advanced content, yet it may even bring about additional mischief by spreading false data and the making of phony news. In this paper, we look at the authenticity of best-in-class Image detections, and that it is so hard to identify them - either consequently or by people. Specifically, we center on Deep Fakes, copy-move, splicing, resembling and statistical. As noticeable delegates for image categorization. Traditional image forensics techniques are usually not well suited to blur images due to the compression that strongly degrades the data. Thus, this paper follows a deep learning approach and presents two networks, both with a low number of layers to focus on the macroscopic properties of images. We make the greater part a million controlled images individually for each approach. The subsequent freely accessible dataset is at any rate a request for greatness bigger than similar other options and it empowers us to prepare information driven phony locators in an administered manner. We will show that the utilization of extra space explicit learning improves imitation identification to an exceptional precision.

Keywords: Deep Fakes, copy-move, splicing

1. Introduction

From the early days, an image has normally been accepted as proof of the amount of the depicted event. Computer becoming more customary in business and other fields, accepting digital image as an official document has become a common practice. The accessibility of low-cost hardware and software tools makes it easy to create, alter, and manipulate digital images with no understandable traces of having been subjected to any of these operations. As a result, we are hastily reaching a situation where one can no longer take the reliability and validity of digital images for granted. This trend undermines the integrity of digital images presented as evidence in a court of law, as news items, as part of a medical records or as financial documents since it may no longer be possible to discriminate whether a given digital image is original or a modified version or even a depiction of a real-life occurrences and objects. Digital image forgery is a mounting problem in criminal cases and in public courses.

Alphanumeric image imitation detection techniques are classified two types of active and passive techniques.

In the active approach, the digital image requires some pre-processing such as watermark embedding or signature generation at the time of creating the image, which would limit their application in practice. Moreover, there are millions of digital images on the internet without a digital signature or watermark. In such a scenario, active approach could not be used to find the verification of the image. Any digital signature generated or watermark embedded in advance is not needed in the passive technology.

There are three techniques widely used to manipulate digital images 1) Tampering – tampering is a manipulation of an image to achieve a specific result. 2) Splicing (Compositing) - A common form of photographic manipulation in which the digital splicing of two or more images into a single composite 3) Cloning (Copy-Move)

1.1 Deep fakes

Dee fakes are media that take a person in an existing image or video and replace them with someone else's similarity using artificial neural networks. They often unite and overlay existing media onto source media using machine learning techniques known as auto encoders and generative adversarial networks.

Deep fakes have garnered prevalent consideration for their uses in celebrity pornographic videos, revenge porn, fake news, hoaxes, and financial fraud. This has elicited responses from both industry and government to detect and limit their use

1.2 Image Splicing

Image Splicing is defined as a paste-up produced by sticking together photographic images. This technique for making forgery images is more destructive than image retouching. Image splicing is essentially simple process and can be done as crops and pastes regions from the same or separate sources.

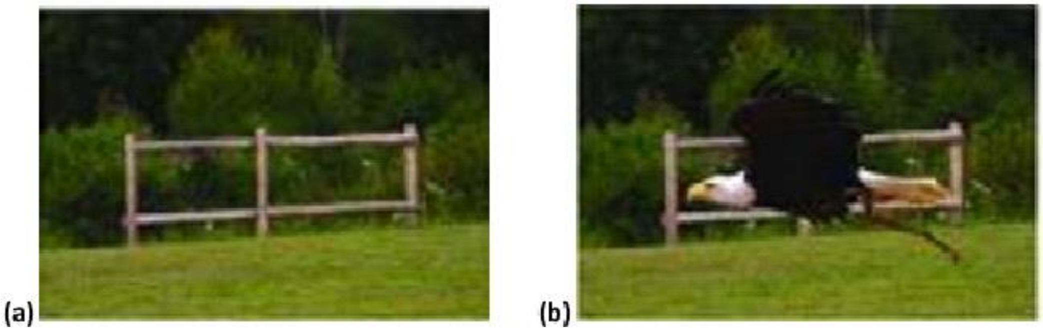


Fig. 1 Image Splicing

Fig. 1 Above shows how to create forge Image; by copying a spliced portion from the source image into a target image. The left picture is the base image and the right one is the spliced image as in that case some cropped image is pasted over the base image and new image is generated

1.3 Copy and Move Image

In a Copy-Move counterfeit, a part of the image itself is copied and pasted into another part of the same image. we can make an object vanish from the image by casing it with a segment copied from added portion of alphanumeric image.

.

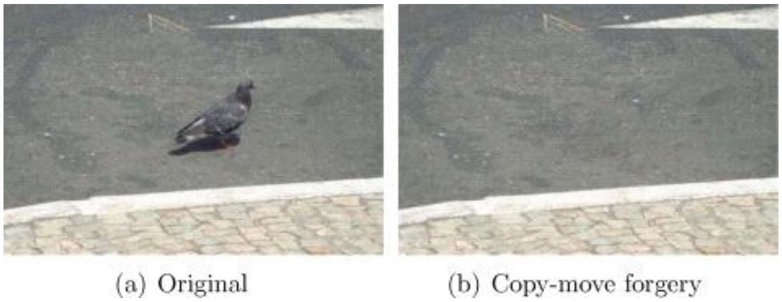


Fig.2 Copy and move Image

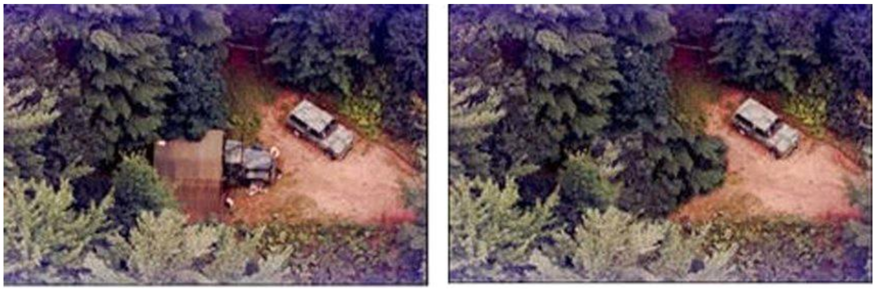


Fig.3 Counterfeit test image of a ‘car’ and its unique form

Examples of the Copy-Move forgery are given in Fig.2. Fig.2 is an obvious counterfeit that was created only for testing. In Fig., we can see a less counterfeit in which a pigeon become disappear (compare the forged image with its original.

In Fig.3, we will see a less counterfeit image in which a bus is covered with a portion of the foliage left of the truck (on comparing counterfeit image with original image). It's not difficult to find the forged area in image because the original and copied parts of the foliage having a lots of similarity.

2.1 COPY-MOVE FORGERY DETECTION METHOD

Any Copy-Move forgery introduces a correlation between the original image segment and the pasted one.

Because the forgery will likely be saved in the lossy JPEG format and because of a possible use of the retouch tool or other localized image processing tools, the segments may not match exactly but only approximately. Thus, we will try to figure out following requirements for a detection algorithm:

1. The detection algorithm must allow for an approximate match of small image segments
2. It should be approach in a reasonable time while introducing uncommon false positives (i.e., detecting incorrect matching areas).
3. Another natural assumption that should be accepted is that the forged segment will likely be a connected component rather than a collection of very small patches or individual pixels.

In this section, two algorithms for detection of the Copy-Move forgery are developed – one that uses an exact match for detection and one that is based on an approximate match. Before describing the best approach based on approximate block matching that produced the best balance between performance and complexity, two other approaches were investigated –Exhaustive search and Autocorrelation

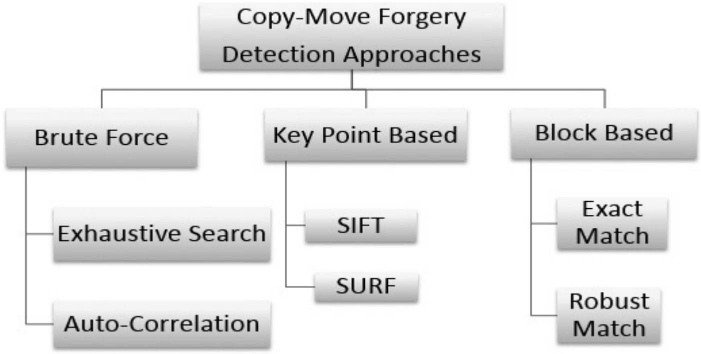


Fig.4. Block diagram of copy-move image forgery detection system

2.2 Exhaustive search

The simplest way to detect a Copy-Move forgery is to use an exhaustive search. In this searching, the image and its circularly shifted version are overlaid looking for closely matching image block.

This technique is simple and pretty good for small-sized images. However, this technique is computationally exclusive and impractical for image of medium-sized although. In this method for an image size, it would take steps, since the comparison and image processing require the order of operations for one shift. One of the other techniques for finding counterfeit is based on autocorrelation. All Copy-Move forgery techniques tries to find a correlation in between original image portion and pasted one.in this approach, there is no large computational complexity and generally not pretty successful to detect forgery.

Generally, the detected image is alienated into overlying blocks. The theme is to find the connected blocks that are being imitative and moved. The copied area consisted of many overlying images blocks. The distance between each duplicate block pair will be same since each portion have been moved with same quantity of shifting. The next challenge will be extracting the features form these small image portions, which would yield to a similar value for imitative blocks. These portions have been vectorized and injected into a matrix and the vectors are lexicographically sorted for later discovery. Computational time depends upon, number of blocks, different sorting techniques and quantity of features in the given image. For example, an image size is, it is divided into overlapping blocks of size b × b. The blocks are represented as vectors of dimensions, and sorted in a lexicographical order (Fig.5). Vectors corresponding to blocks of similar content would be close to each other in the list, so that identical regions could be easily detected

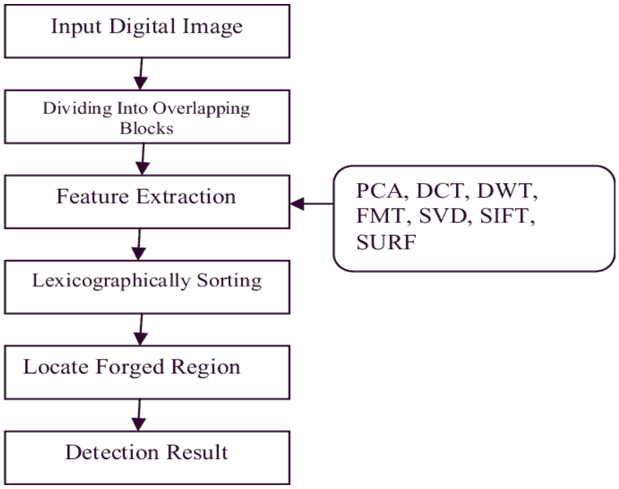
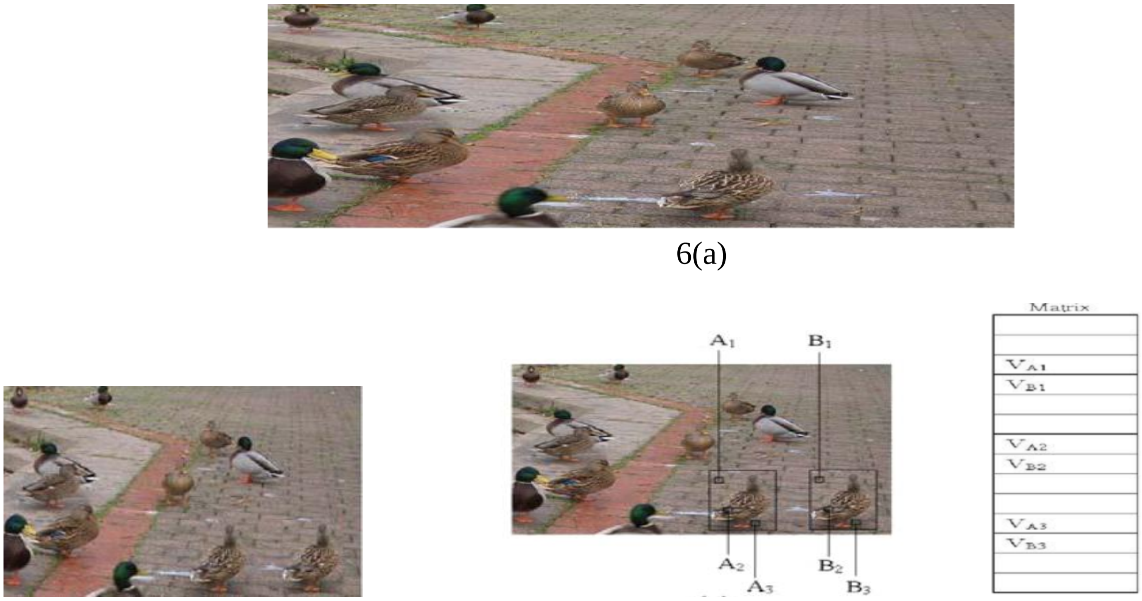

Input Digital Image
Dividing Into Overlapping Blocks
Feature Extraction
PCA, DCT, DWT, FMT, SVD, SIFT, SURF
Lexicographically Sorting
Locate Forged Region
Detection Result

Fig.5 Configuration of a block Copy-Move Digital Image Forgery Detection System

The image given in Figure 6(a) is the original image and Figure 6(b) is the tampered image by Copy-Move Forgery.



6(a)

6(b) 6(c) 6(d)

As shown in Figure 6 (c), the block B1, B2, and block B3 which are copies of blocks A1, A2, and block A3, respectively. Therefore, VA1 =VB1, VA2 =VB2, and VA3 =VB3, where VX denotes the vector corresponding to block X. As shown in sorted list, Figure 6(d), identical vectors are adjacent each other

Fig. 6 (a). original image (b). Counterfeit image (c) Three pairs of identical blocks are marked

(d). Feature vectors corresponding to the divided blocks are sorting in a list

2.2.1 Auto correlation

The autocorrelation of the image x of the size M×N is defined by the formula:

$$r_{k,l} = \sum_{i=1}^{M}\sum_{j=1}^{N} x_{i,j} x_{i+k,j+l},\ i,k = 0,\dots,M-1,\ j,l = 0,\dots,N-1\,.$$

The autocorrelation can be efficiently implemented using the Fourier transform utilizing the fact that r = x* xˆ, where xˆij = xM+1–i, N+1–j, i = 0, …, M–1, j = 0, N–1. So, we have
r = F–1{F (x) F (xˆ)},
where F denotes the Fourier transform.
Idea behind detection depend on autocorrelation is that the original and imitative parts will introduce crests in the autocorrelation for the shifts that correspond to the copied-moved portions. Because natural images contain most of their control in low-frequencies, if the autocorrelation r is calculated directly for image itself, r would have very large crests on image corner and neighbors.

so, we will calculate the autocorrelation by its high-pass filtered version ,it's not going to be calculated directly from image.

For this purpose, we tested : Laplacian edge detector ,Marr edge detector, , Sobel edge detector, and noise extracted using the 3×3 Wiener filter. The best results were obtained by using the 3×3 Marr filter. For example, the minimum size of a copied-moved segment is m, the autocorrelation copy-move detection method consists of the following steps:

1. implement Marr high-pass filter to the tested one alphanumeric image.
2. now we will Compute auto correlation r of a filtered image.
3. Remove half of the autocorrelation (Autocorrelation is symmetric.).
4. Set a = 0 in the neighborhood of two remaining corners of the entire autocorrelation.
5. Find the maximum of a, identify the shift vector, and examine the shift using the exhaustive method.
6. If the detected area is larger than m, finish, else repeat Step 5 with the next maximum of a.

It's a simple technique and not having large complexity in computation, but it fails to detect the forgery if the size of the forged area should be more than 1/4th of linear image dimensions (according to our program). So Exhaustive search and the Autocorrelation method are skipped in favor of another technique that will work comparatively better and faster than previous techniques.

2.3 Detection of Copy-Move Forgery by Block Matching

2.3.1 Exact match

The first algorithm described in this section is for identifying those segments in the image that match exactly. Even though the applicability of this tool is limited, it may still be useful for forensic analysis. This builds a basis for a strong match, given detailed in the next para. Initially, user specifies the minimum size of a section that will be consider for matching. Let us suppose that this segment is a square with P×P pixels. The square is slid by one pixel along the image from the upper left corner right and down to the lower right corner. For each position of the P×P block, the pixel values from the block are extracted by columns into a row of a two-dimensional array A with B^2 columns and (A–P+1) (B–P+1) rows. Each row corresponds to one position of the sliding block. Two identical rows in the matrix A correspond to two identical PXP blocks. To identify the identical rows, the rows of the matrix A are lexicographically ordered (as PXP integer tuples). This can be done in ABlog2(AB) steps. The matching rows are easily searched by going through all ab rows of the ordered matrix a and looking for two consecutive rows that are identical.

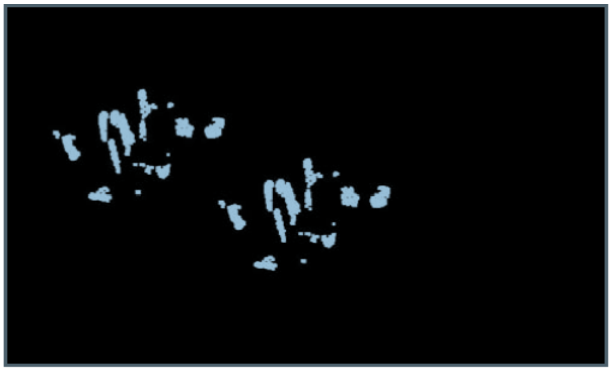


Figure 7 Block Match Copy-Detection forgery algorithm

(The exact match mode with block size P=2).

The matching blocks found in the Bitmap image of CAR (Figure 3) for P=8 is shown in Figure. The blocks form an irregular pattern that closely matches the copied-and-moved foliage. The fact that the blocks from several disconnected pieces instead of one connected segment indicates that the person who did the forgery has probably used a retouch tool on the pasted segment to cover the traces of the forgery. It’s important to remind that if the counterfeit image had been saved as jpeg, vast majority of identical blocks would have disappeared because the match would become only approximate and not exact.

In the next , algorithm for the robust match and its performance should be computed

2.3.2 Robust match

The idea for the robust match detection is similar to the exact match except we do not order and match the pixel representation of the blocks but their robust representation that consists of quantized DCT CONSTANTS. The quantization steps are calculated from a user-specified parameter Q. This parameter is equivalent to the quality factor in JPEG compression, i.e., the Factor determines the quantization steps for DCT transform CONSTANTS. Because higher values of the Q-factor lead to finer quantization, the blocks must match more closely in order to be identified as similar. Lower values of the Q-factor produce more matching blocks, possibly some false matches.

The detection begins in the same way as in the exact match case. The image is scanned from the upper left corner to the lower right corner while sliding a B×B block. For each block, the DCT transform is calculated, the DCT CONSTANTS are quantized and stored as one row in the matrix A. The matrix will have (M– B+1) (N–B+1) rows and B×B columns as for the exact match case. The rows of A are lexicographically sorted as before. The remainder of the procedure, however, is different. Because quantized values of DCT CONSTANTS for each block are now being compared instead of the pixel representation, the algorithm might find too many matching blocks (false matches). Thus, the algorithm also looks at the mutual positions of each matching block pair and outputs a specific block pair only if there are many other matching pairs in the same mutual position (they have the same shift vector). Towards this goal, if two consecutive rows of the sorted matrix A are found, the algorithm stores the positions of the matching blocks in a separate list (for example, the coordinates of the upper left pixel of a block can be taken as its position) and increments a shift-vector counter C. Formally, let (i1, i2) and (j1, j2) be the positions of the two matching blocks. The shift vector s between the two matching blocks is calculated as

$$s = (s1, s2) = (i1 – j1, i2 – j2)$$

Because the shift vectors –s and s correspond to the same shift, the shift vectors s are normalized, if necessary, by multiplying by –1 so that $s1 \geq 0$. For each matching pair of blocks, we increment the normalized shift vector counter C by one:

$$C (s1, s2) = C(s1 , s2) + 1 .$$

The shift vectors are calculated and the counter C incremented for each pair of consecutive matching rows in the sorted matrix A. The shift vector C is initialized to zero before the algorithm starts. At the end of the matching process, the counter C indicates the frequencies with which different normalized shift vectors occur. Then the algorithm finds all normalized shift vectors s(1),s(2), …, s(K), whose occurrence exceeds a user-specified threshold T: C(s(r)) > T for all r = 1, …, K. For all normalized shift vectors, the matching blocks that contributed to that specific shift vector are colored with the same color and thus identified as segments that might have been copied and moved.

The value of the threshold T is related to the size of the smallest segment that can be identified by the algorithm. Larger values may cause the algorithm to miss some not-so-closely matching blocks, while too small a value of T may introduce too many false matches. We repeat that the Q factor controls the sensitivity of the algorithm to the degree of matching between blocks, while the block size B and threshold T control the minimal size of the segment that can be detected.

For the robust match, we have decided to use a larger block size, B=16, to prevent too many false matches (larger blocks have larger variability in DCT CONSTANTS). However, this larger block size means that a 16×16 quantization matrix must be used instead of simply using the standard quantization matrix of JPEG. We have found out from experiments that all AC DCT CONSTANTS for 16×16 blocks are on average 2.5 times larger than for 8×8 blocks and the DC term is twice as big. Thus, the quantization matrix (for the Q-factor Q) that is used for quantizing the DCT CONSTANTS in each 16×16 block has the following form

$$Q_{16} = \begin{pmatrix} Q'_8 & 2.5q_{18}I \\ 2.5q_{81}I & 2.5q_{88}I \end{pmatrix}, \text{ where } Q'_8 = \begin{pmatrix} 2q_{00} & 2.5q_{12} & \dots & 2.5q_{18} \\ 2.5q_{21} & 2.5q_{22} & \dots & 2.5q_{28} \\ \dots & \dots & \dots & \dots \\ 2.5q_{81} & 2.5q_{82} & \dots & 2.5q_{88} \end{pmatrix}$$

and qij is the standard JPEG quantization matrix with quality factor Q and I is an 8×8 unit matrix (all elements equal to 1). We acknowledge that this form is rather ad hoc, but because the matrix gave very good performance in practical tests and because small changes to the matrix influence the results very little, we did not investigate the selection of the quantization matrix further. Note regarding color images: In both Exact and Robust Match, if the analyzed image is a color image, it is first converted to a grayscale image using the standard formula I = 0.299 R + 0.587 G + 0.114 B, before proceeding with further analysis.

3. Conclusion

As Copy-Move forgeries have become popular, the importance of forgery detection is much increased. Although many Copy-Move Forgery detection techniques have been proposed and have shown significant promise, robust forgery detection is still difficult. There are at least three major challenges: tampered images with compression, tampered images with noise, and tampered images with rotation. In this paper we reviewed several papers to know the recent development in the field of Copy-Move digital image forgery detection. Sophisticated tools and advanced manipulation techniques have made forgery detection a challenging one. Digital image forensic is still a growing area and lot of research needed to be done.

References

H.T. Sencar, and N.Memon, “Overview of State-of-the Art in Digital image Forensics”, World Scientific Press, 2008

H . Farid, “A Survey of image forgery detection”, IEEE Signal Processing Magazine, Vol. pp. 16-25, 2009

B.L.Shivakumar and S.Santhosh Baboo, “Digital Image Forgery Detection”, SAJOSPS, Vol. 10(2), pp. 116-119, 2010

Lou Weigi, Qu Zhenhua, Pan Feng, and Herang Jiwu, “ Survey of Passive Technology for Digital Image Forensics”, Frontiers of Computer Science in China, Vol. 1(2), pp. 166-179, May 2007

J. Fridrich, “Methods for "Methods for Tamper Detection in Digital Images", Proc. ACM Workshop on Multimedia and Security, Orlando, FL, October 30−31, 1999, pp.19−23.

S. Saic, J. Flusser, B. Zitová, and J. Lukáš, “Methods for Detection of Additional
Manipulations with Digital Images”, Research Report, Project RN19992001003
"Detection of Deliberate Changes in Digital Images", ÚTIA AV ČR, Prague, December
1999 (partially in Czech).

J. Lukáš, “Digital Image Authentication“, Workshop of Czech Technical University 2001,Prague, Czech Republic, February 2001

Nizza, M., Lyons, P.J.: In an iranian image, a missile too many. In: The Lede, The New York Times News Blog (2008) http://thelede.blogs.nytimes.com/ 2008/07/10/in-an iranian-image-a-missile-too-many/.

Fridrich, D. Soukal, and J. Lukás, “Detection of copy move forgery in digital images,” in Proc. Digital Forensic Research Workshop, Aug. 2003.

A. C. Popescu and H. Farid, “Exposing Digital Forgeries by Detecting Duplicated Image Regions,” Technical Report, TR2004-515, Department of Computer Science, Dartmouth

G. Li, Q. Wu, D. Tu, and S. Sun, “A Sorted Neighborhood Approach for Detecting Duplicated Regions in Image Forgeries based on DWT and SVD,” in Proceedings of IEEE International Conference on Multimedia and Expo, Beijing China, July 2-5, 2007, pp. 1750-1753.

W. Luo, J. Huang, and G. Qiu, “Robust Detection of Region Duplication Forgery in Digital Image,” in Proceedings of the 18th International Conference on Pattern Recognition, Vol. 4, 2006, pp. 746-749.

A. N. Myna, M. G. Venkateshmurthy, and C. G. Patil, “Detection of Region Duplication Forgery in Digital Images Using Wavelets and Log-Polar Mapping,” in Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007), Vol. 3, pp. 371-377, 2007.

H . Huang, W. Guo, and Y. Zhang, “Detection of Copy-Move Forgery in Digital Images Using SIFT Algorithm,” in Proceedings of IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application, Vol. 2, pp. 272-276, 2008.

Jing Zhang, Zhanlei Feng and Yuting Su, “A New Approach for Detecting Copy-Move Forgery in Digital Images”, in: IEEE Singapore International Conference on Communication Systems, Guangzhou, China, pp. 362-366, 2008

Hwei-Jen Lin, Chun-Wei Wang, Yang-Ta Kao, “Fast Copy-Move Forgery Detection”, in WSEAS Transaction on Signal Processing, Vol 5(5), pp. 188-197, May 2009.

Xunyu Pan and Siwei Lyu, “Detecting Image Region Duplication Using SIFT Features”, in: International Conference on Acoustics, Speech, and Signal Processing, Dallas, TX, 2010

Seung-Jin Ryu, Min-Jeong Lee and Heung-Kyu Lee, “Detection of Copy-Rotate-Move Forgery using Zernike Moments”, in: 12th International Workshop on Information Hiding, Calgary, Alberta, Canada, 2010

ABOUT THE AUTHORS

Gavendra Singh (PhD) working as an Assistant Professor in the department of Software Engineering, in College of Computing and Informatics, Haramaya University, He has published research papers in machine learning, image processing and artificial intelligence.
email: yashgaven11@gmail.com

Faizur Rashid (PhD) is working as Assistant Professor in Computer Science Department in college of computing and Informatics, Haramaya University since 2012, He has published various research papers in international Journals, His research area includes from Artificial Intelligence, Machine Learning, Computer Vision and Image Processing, NLP.
email: faizurrashid@hotmail.com

Mr. Afendi Abdi Mohammed is working as Lecturer and Associate Dean in College of Computing and Informatics, Haramaya University. His research area includes machine learning and NLP.
email: afe2003@gmail.com