



## **Copy Move Image Forgery Detection with Exact Match Block Based Technique**

**PRIYANKA\* and DERMINDER SINGH**

Department of Electrical Engineering and Information Technology,  
Punjab Agricultural University, Ludhiana, Punjab, India.

### **Abstract**

Digital images are a momentous part of today's digital communication. It is very easy to manipulate digital images for hiding some useful information by image rendering tools such as Adobe Photoshop, Microsoft Paint etc. The common image forgery which is easy to carry out is copy-move in which some part of an image is copied and pasted on another part of the same image to hide the important information. In this paper we propose an algorithm to spot the copy-move forgery based on exact match block based technique. The algorithm works by matching the regions in image that are equivalent by matching the small blocks of size  $b \times b$ . The program is tested for 45 images of mixed image file formats by considering block sizes 2, 4, 6, 8, 10, 12, 14, and 16. It is observed from the experimental results that the proposed algorithm can detect copy-move image forgery in TIF, BMP and PNG image formats only. Results reveal that as the block size increases, execution time (time taken by CPU to display output) also increases but the number of detected forged images increases till block size 10 and attains saturation thereafter. Consequently block size should be set to 10 for getting good results in terms of less execution time.



### **Article History**

Received: 08 July 2019

Accepted: 29 July 2019

### **Keywords**

Block Size;  
Copy Move Image Forgery;  
Exact Match Block Based Method;  
False Matches;  
Image Forgery;  
Lexicographic Sorting.

### **Introduction**


In today's digital world, images are a significant part of digital communication. An image can define a situation better than words. Digital images are used in medical science, forensic investigation, journalism, marketing, agriculture and most extensively in social

networking websites such as Instagram, Facebook, and Twitter etc. From the time when photography was invented, organisations and individuals have often searched many ways to modify images in order to mislead its viewer. Initially it was equitably a difficult task, as it required many hours of effort

**CONTACT** Priyanka ✉ priyanka.arora3110@gmail.com 📍 Department of Electrical Engineering and Information Technology, Punjab Agricultural University, Ludhiana, Punjab, India.



© 2019 The Author(s). Published by Oriental Scientific Publishing Company

This is an  Open Access article licensed under a Creative Commons license: Attribution 4.0 International (CC-BY).

Doi: <http://dx.doi.org/10.13005/ojcs12.03.07>

by a professional expert. However, with the advent of digital technology it has become easy to modify images and achieve professional results as reported by Sharma (2014). Any person who does not have enough knowledge about the background of digital images can alter the foreground visual of an image by using user friendly image processing software and it can change the intact meaning of the image. It is of no harm if done for improving their pictures to post on social networking websites. But it is an offence when changes are made on an image which is a proof of a criminal investigation. This is called digital image forgery. Keeping in mind the forensic reasons it is essential to spot forgery. The type of image forgery which is easiest to do is copy move image forgery in which a section of an image is cloned and pasted to some another section of the same image as shown in Figure 1.

In this paper we propose an algorithm to detect copy move forgery which matches small regions in image of size  $b \times b$  and declares those regions as forged which match exactly.

#### Literature Reviewed

In the past few years researchers have developed several techniques to spot image forgery.

The main types of forgery are image splicing and image cloning.<sup>16</sup> These methods work on the main idea that there is a correlation between the copied and moved region. The first method is exhaustive search in which all the pixels are matched to detect the forgery.<sup>2</sup> Next is Key-Point based in which SIFT or SURF features are computed for key-points for forgery detection.<sup>10</sup> Then in block based method the image is divided into small sized blocks, then these are matched for forgery detection in exact match.<sup>2</sup> Functions such as DCT<sup>1,5,9</sup>, DFrWT<sup>4</sup>, SVD<sup>7</sup>, PCA<sup>7</sup> is applied on the divided blocks, and then these are matched for forgery detection.

#### Copy Move Digital Image Forgery

It is a very common type of forgery in which a segment of image itself is imitated and pasted on another segment. It is done in order to hide some information present on image as it is easy to copy a part of image and paste into another position of the same image using user friendly image processing tools. There are two types of information present in an image:- the background and the main objects. The background information such as greenery, stones, sky, ground, water, buildings and fabric are irregular surfaces suitable for this kind of forgery as the area copied from this context gets merged in the image



Fig. 1: Original Image (left) Forged Image (right)



Fig. 2: Original Image (left) and copy-move forged image (right)

in such a way that it is not visible by the human eye.<sup>16</sup> Figure 2 shows a street lamp is replicated and inserted at another position near the tree, which looks very realistic.

### Copy Move Forgery Detection Approaches

The methods explored by researchers are shown in Figure 3 are illuminated below:

**Brute Force** is a basic copy-move forgery detection type using Exhaustive Search and Autocorrelation explained as:

**Exhaustive Search** is an easy-going approach of detecting copy-move forgery. A digital image is a representation of a real image as a set of numbers called the picture elements commonly called pixels. Pixels can be stored and handled by a digital computer. For each pixel, the imaging device records a number that describe some property of this pixel such as intensity of light or its colour. The idea is to match each pixel value with other pixel values, starting from top left corner of image to bottom right corner and mark the duplicated pixel. Exhaustive search uses circularly shifted versions of forged image to match with other parts. It reduces the computational complexity as a pixel value is matched twice with other pixel values, so half of the comparisons are reduced. But there are two limitations of this technique:

An image of size  $400 \times 400$  will require  $400!$  ( $6.4034 \times 10^{868}$ ) contrasts, which is certainly a very large amount resulting in a high complexity.

A grayscale image uses pixel values between 0-255. According to pigeonhole principle it is certain that almost half of the pixel values will be repeated.

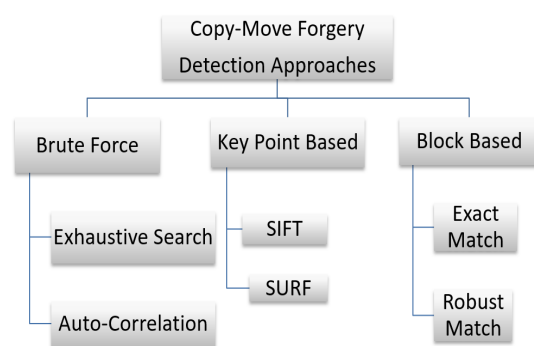
**Autocorrelation** works on the logic that the copy-moved i.e. the shifted parts show peaks in autocorrelation, which are an indication of forgery.

**Key Point Based** clone detection method converts the color space of image if required and uses either of two methods SIFT and SURF to extract unique features and matches them to detect forgery. The process is shown in Figure 4:

**Block Based** copy-move forgery detection method uses two types of matches Exact Match and Robust Match as described below:

### Exact Match

This method is used to find those image segments which match exactly. As shown in Figure 6, a square block of size  $b$  is moved over the image of size  $M \times N$ , starting from top left corner to bottom right corner, matrices of size  $b \times b$  are taken out and stored in a two dimensional array  $A$  with  $(M-b+1) \times (N-b+1)$  rows and  $b \times b$  columns. The two indistinguishable rows in array  $A$  relate to two matching image segments. Then, instead of matching each row with other row, all the rows are lexicographically sorted. As a result, the rows having similar pixel values come closer. Therefore the task reduces to matching a row with its neighbour only, which reduces the computational complexity of matching steps.<sup>5</sup> This can be done



**Fig. 3: Copy-Move Forgery Detection Approaches**

**Table 1: Average Results**

Block Size	Execution Time	No. of Images Correctly Detected
2	11.7055	32
4	18.32853	37
6	32.96058	41
8	55.12518	42
10	77.49838	45
12	104.8614	45
14	140.5369	45
16	176.4153	45

only in  $MN \log_2(MN)$  number of steps. The matching segments are highlighted.

#### Robust Match

It works on the idea similar to exact match. First, the

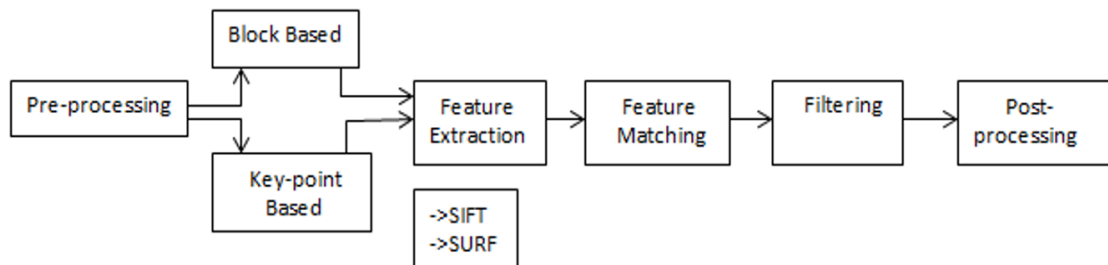


Fig. 4: Steps of Key-point based clone detection method

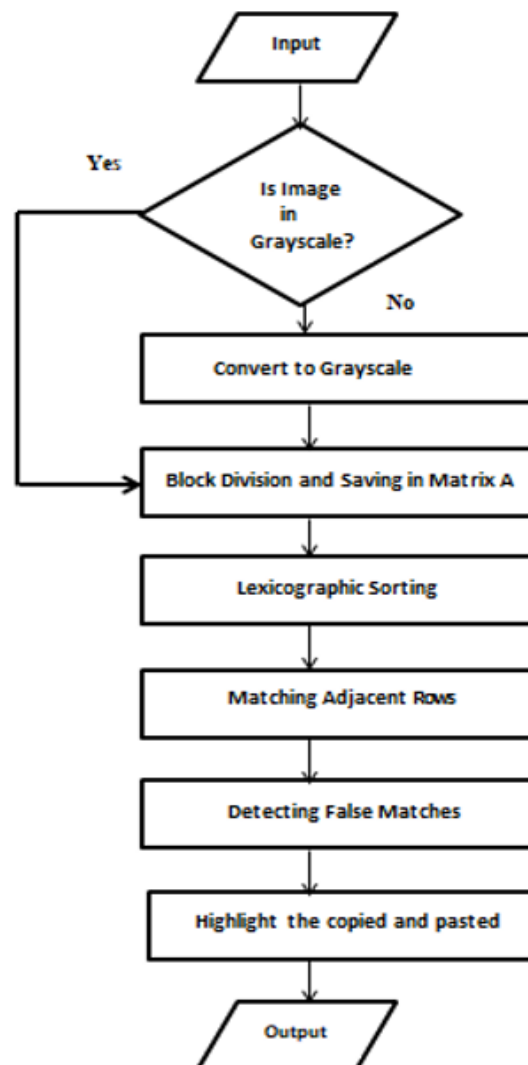


Fig. 5: Flowchart of Proposed Algorithm

blocks are extracted as in exact match. But instead of matching the exact pixel values of blocks, a function is applied on blocks. Next, unique features are extracted from each block. Then these features are matched in order to detect forgery.

### Material and Methodology

#### Material

Many researchers use MATLAB to implement their research.<sup>1,7,11,12,13,14,15</sup> However, the proposed algorithm is implemented by converting it to a program in Octave language as it's community edition is open source and its scripts are compatible with MATLAB scripts.

#### Methodology

The proposed algorithm is based on exact match block based technique. Initial tests have shown that it takes much more time in matching all the possible regions in an image. Thus some assumptions are formulated for the detection algorithm<sup>2</sup>:

- It must match the small image fragments.
- It must work in a practical time with less number of images that have been falsely detected as forged.

- The forged part must be connected instead of individual pixels and no post-processing should be done on the image.

The steps involved to detect copy move forgery are shown in Figure 3 and described as:

#### Pre-Processing

The input image is converted into grayscale as grayscale is easy to handle. A standard formula is used to convert RGB image into grayscale image which is  $I = 0.299 R + 0.587 G + 0.114 B$  where R, G, B are the three frames a coloured input image and I is the resulting image.<sup>4,11,15</sup>

#### Block Division

In square block partitioning, the color space converted image of size  $M \times N$  is divided into square blocks of size  $b$  by overlapping a window of size  $b$ . Starting from top left corner to the bottom right corner, extraction of total  $(M-b+1) \times (N-b+1)$  blocks is done as shown in Figure 6. However, the window size must be chosen cautiously, because if a window size larger than forged region is chosen, it will not be able to detect the possible forgery. The obtained matrices are stored in a array A as shown

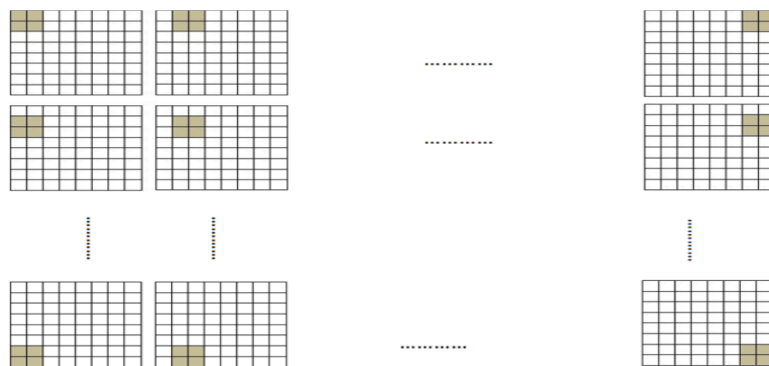


Fig. 6: Block Division in Block Based Copy Move Forgery Detection Method (b=2)

$$A = \begin{bmatrix} V1 \\ \vdots \\ V(M-B+1)(N-B+1) \end{bmatrix}$$

Fig. 7: Storage of Matrices in Matrix A

below. As there are a total of  $Nb = (M-b+1) \times (N-b+1)$  blocks, so there are total  $Nb$  rows and  $b \times b$  columns in matrix A. As shown in Figure 7, the rows be stored as  $V_1, V_2, \dots, V_{(M-b+1) \times (N-b+1)}$ .<sup>5,7</sup>

### Lexicographic Sorting

Next, the rows are compared by presuming that the copied regions would have the same rows. However, if a row is matched with rest of the rows, the computation cost will be significantly high, especially when the size of the image is large. In order to reduce the time of matching, the similar rows will be stored into the neighbour rows by lexicographical sorting. In this way, similar blocks will locate at the neighbouring rows and matching can be achieved in less time. It can be better visualized as shown in Figure 8.<sup>3</sup>

Figure 8 (a) shows the original image and Figure 8 (b) shows the forged image. As shown in Figure 8 (c) the blocks  $P_1$ ,  $P_2$  and  $P_3$  are copies of blocks  $Q_1$ ,  $Q_2$  and  $Q_3$  respectively. It is assumed that they have the feature vectors  $VP_1$ ,  $VP_2$ ,  $VP_3$ ,  $VQ_1$ ,  $VQ_2$  and  $VQ_3$  where  $V_i$  denotes the vectors corresponding to block  $B_i$ . Consequently  $VP_1=VQ_1$ ,  $VP_2=VQ_2$  and  $VP_3=VQ_3$ . When the features are stored in matrix, they are stored in unsorted manner as shown in Figure 8 (d)

and after lexicographic sorting they get stored as in Figure 8 (e).

### Block Matching

As the rows correspond to the blocks of image, for block matching the sorted rows are matched. There is no need to match all the rows with each other because due to lexicographic sorting, the similar rows are next to each other. So starting from the first row, the consecutive rows are matched and saved for further processing.

### Block Filtering

There can be some parts in an image which are repeating such as grass, background etc. The forgery detection algorithm will declare these parts as forged. To avoid this, the algorithm calculates mutual position of matching pairs of blocks and outputs that block pair if there are many other block pairs with same mutual position. So if two matching rows are found in array A, the algorithm takes the position as the co-ordinates of upper left pixel of block and stores in a separate list. Let the positions of two matching blocks be  $(x_1, y_1)$  and  $(x_2, y_2)$ . Then shift vector is calculated as  $s=(x_1-x_2, y_1-y_2)$ . A counter C is initialized to zero and is incremented if there are

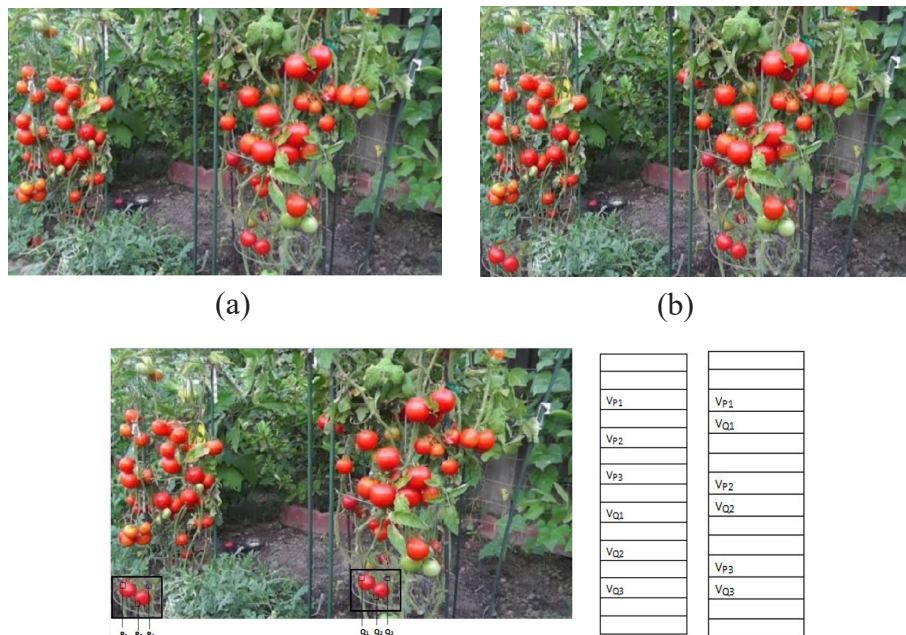


Fig. 8: (a) Original Image (b) Forged Image (c) Blocks of Copied and Pasted Part (d) Unsorted Matrix (e) Sorted Matrix



many other shift vectors of matching blocks which are similar. All the shift vectors  $s_1, s_2, s_3, \dots, s_r$  are calculated. The counter specifies the frequency with which the shift vectors occur. The rows which have maximum number of same shift vector values are stored and then the blocks corresponding to stored rows are highlighted.<sup>2</sup>

### Experimental Results

Dataset of total 45 downloaded images is used for testing the program. The size (height\*width) of tested images varies between 150\*200 to 338\*149. The downloaded images were in JPG image format. However the program did not gave satisfactory results for images with this format. Then all the sample images were converted into TIF, GIF, BMP and PNG image file formats. The program gave satisfactory results for only TIF, BMP and PNG images file formats. The result of the forged image in TIF format when tested by proposed algorithm is obtained as shown in Figure 9 by taking  $b=8$ .

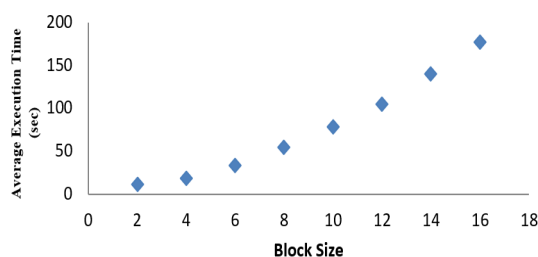
Then the images with these TIF, BMP and PNG formats are forged to test the algorithm by using Microsoft Paint. The execution time i.e. time taken by CPU to execute the program for each image is

calculated and it is noted that whether the test image is correctly detected as forged or not. Then the average execution time and total number of correctly detected images are calculated and the results are shown in Table 1. The results can be better visualized as shown in Graph: 1 and Graph: 2.

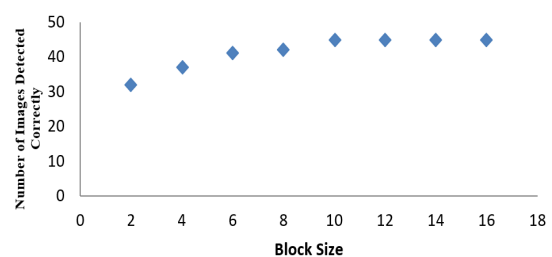
It can be clearly seen from Graph 1 that if the block size increases, there is an increase in time taken by proposed algorithm to display the result, because the size of the extracted block finalizes the number of columns. If the size of block is  $b \times b$ , then the numbers of columns in array are  $b^2$ . Thus, as the block size increases there is an increase in number of columns. As the algorithm compares the values of columns of two adjacent rows for similarity, if the block size will increase, there will be increase in number of columns to be compared. Consequently the time taken will increase. But according to Graph 2 it is also seen that, as the block size increases, the number of images correctly detected by the proposed algorithm also increases up to block size 10 and attains a saturation value after that. Thus, it is concluded that the block size must be taken 10 for getting better results in terms of less execution time and more number of correctly detected images.



**Fig. 9: Output of Forged Image Tested by Proposed Algorithm**



**Graph 1: Average Execution Time vs varying Block size**



**Graph 2: Number of Images Correctly Detected vs varying Block size**

### Conclusion

In the past few years, copy move forgery detection has become an emerging area in terms of research. Researchers have proposed different techniques to detect this kind of forgery. But it is difficult for a new researcher to start from scratch. Thus we have proposed an algorithm based on exact match. In experimental results it is seen that this algorithm worked well on TIF, BMP and PNG image file formats as these are lossless file formats. Also the average execution time and number of correctly detected images increase with the increase in block size i.e. as we increase the block size the accuracy increases but the execution time also increases. Also the graph of correctly detected images attains a saturation value after block size 10 and execution

time increases with increasing block size. Thus block size should be taken 10 for getting better results in terms of less execution time and more number of correctly detected images. This work will greatly help the researchers who are new in this field. The study can be further extended by applying robust block based technique and test for lossy image file formats.

### Acknowledgement

Priyanka, Derminder Singh, Department of Electrical Engineering and Information Technology, Punjab Agricultural University, Ludhiana, Punjab, India

### Conflict of Interest

No conflict

### References

1. Ashima Gupta, Nisheeth Saxena and S.K Vasistha: "Detecting Copy Move Forgery using DCT", *International Journal of Scientific and Research Publications*, Volume 3, Issue 5, May 2013.
2. Jessica Fridrich, David Soukal, and Jan Lukas: "Detection of Copy-Move Forgery in Digital Images", in *Proceedings of Digital Forensic Research Workshop*, August 2003.
3. B. L. Shivakumar and S. S. Baboo: "Detecting copy-move forgery in digital images: a survey and analysis of current methods", *Global Journal of Computer Science and Technology*, vol. 10, no. 7, pp. 61–65, 2010.
4. Amanjot Kaur Lamba, Neeru Jindal and Sanjay Sharma: "Digital image copy-move forgery detection based on discrete fractional wavelet Transform", *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 26, pp. 1261-1277, 2018.
5. Rohini.A.Maind, Alka Khade, D.K.Chitre: "Image Copy Move Forgery Detection using Block Representing Method", *International Journal of Soft Computing and Engineering*, Volume-4, Issue-2, pp. 49-53, May 2014.
6. Girish R. Talmale and Yogesh Malode, "Study Of Different Techniques Of Image Forgery Detection", *International Journal of Advanced Research in Computer Science*, Volume 4, No. 1, January 2013 pp.8-13.
7. Kavya Sharma: "Computationally Efficient Copy-Move Image Forgery Detection Based on DCT and SVD", *Advanced Research in Electrical and Electronic Engineering*, Volume 1, Number 3 (2014) pp.76-81.
8. A.C.Popescu and H.Farid: "Exposing Digital Forgeries by Detecting Copied Image Regions", Dartmouth College, 2004.
9. Nathalie Diane Wandji, Sun Xingming, Moise Fah Kue, "Detection of copy-move forgery in digital images based on DCT", *Journal of Computer Science*, vol. 10, pp. 295–302, 2013.
10. Gagandeep Kaur And Manoj Kumar, "Study of Various Copy Move Forgery Attack Detection Techniques in Digital Images", *International Journal of Research in Computer Applications and Robotics*, Vol.3, Issue 9, Pg.: 30-34, September 2015.
11. S. Subah, S. Derminder and C. Sanjeev, "An interactive computer vision system for tree ring analysis", *Current Science*, Vol. 112, Issue 6, March 2017, pp. 1262-1265.
12. Nikita Singla and Derminder Singh, "A Soft Approach to Estimate Woody Volume of a Live



- Tree", *Oriental Journal of Computer Science and Technology*, Volume 10 – No.3, 2017, pp 618-623.
13. Shweta and Derminder Singh, "Computer Aided Leaf Morphometric Approach For The Identification of Regional Plant Species ", *Environment and Ecology*, Vol.34, No.3C, pp. 1556-1561.
  14. Sandhya, Mahesh Kumar and Derminder Singh, "Engineering Characterization Of Tomato Using Image Processing", , Vol. 55, issue 3, September 2018, pp. 510-515.
  15. Sukhvir Kaur and Derminder Singh, "Geometric Feature Extraction of Selected Rice Grains using Image Processing Techniques", *International Journal of Computer Applications* (0975 – 8887) Volume 124 – No.8, August 2015, pp 41-46.
  16. A. Rocha, W. Scheirer, T. E. Boult, and S. Goldenstein, "Vision of the unseen: Current trends and challenges in digital image and video forensics," *ACM Computing Surveys*, vol. 43, no. 4, 2011.