International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT

ELSEVIER SSRN

IJIEMR Transactions, online available on 9th Jul 2019. Link :http://www.ijiemr.org/downloads.php?vol=Volume-08&issue=ISSUE-07

Title: ANALYSIS OF IMAGE DETECTION

Volume 08, Issue 07, Pages: 65–73.

Paper Authors

PRANEEL KUMAR PERURU

JNTUA College of Engineering, Ananthapuramu.

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

ANALYSIS OF IMAGE DETECTION

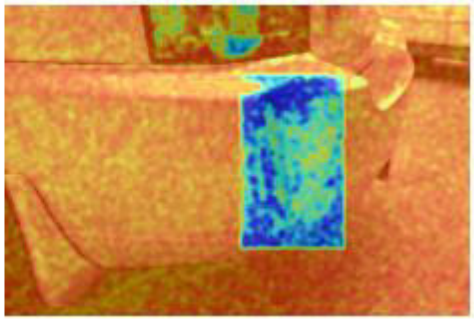
PRANEEL KUMAR PERURU
Research Scholar, Department of CSE, JNTUA College of Engineering, Ananthapuramu.

ABSTRACT
Modern technology makes it easier to manipulate images. It is important to detect image forgery effectively since it can be malicious and has severe consequences. Different algorithms have been designed to detect various image tampering operations. We construct a dataset from the available raw images and benchmark the algorithms. Moreover, we conduct preliminary fusion according the output maps of the algorithms.

1. INTRODUCTION
As growing of media communication and video on demand is desired, image data compression has received an increasing interest. The main purpose of image compression is to gain a very low bit rate and achieve a high visual quality of decompressed images. Image compression are used all fields of media communication such as multimedia, medical image recognition, digital image processing. The fundamental techniques of video compression are based on the schemes of still gray level image compression and colored image compression. Images can be manipulated in various ways. Different touch-ups and image manipulation techniques are applied to augment or enhance a given image. For instance, mobile applications such as Instagram are very popular where users apply “filters” to improve the presentation of their images. Furthermore, images are regularly re-sized and recompressed, such that they can be more easily exchanged over the Internet due to the proliferation of cloud-based photo sharing and editing websites like Flickr and Picasa, spurred by the social media applications like WhatsApp, Instagram and Snapchat. These manipulations are typically not recognized as Image Tampering as the intent to manipulate the information content of the images is minimal. However, there exists many instances where nefarious intent is the sole purpose of manipulating images, such as manipulating the dollar amount on receipt images. Figure 1 illustrates an example of a forged image: the left picture is a manipulated picture showing a dent on a car, while the testing result on the right shows that it is highly possible that the dent is not authentic.The proliferation of image processing software, such as Photoshop and GIMP, provides the necessary tools to

(a) The manipulated image

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org



(b) The likelihood map

Figure 1: Manipulated image of a car and testing result achieve malicious manipulation of images with ease.

This makes the reliable detection of tampering instances an important task. The process of detecting such manipulations is termed Image Forensics. An image forensics algorithm shall output information indicating whether the image hasBeen tampered with, as well as more importantly, identify the portions of the image that have been altered.Achieve malicious manipulation of images with ease. This makes the reliable detection of tampering instances an important task. The process of detecting such manipulations is termed Image Forensics. An image forensics algorithm shall output information indicating whether the image has been tampered with, as well as more importantly, identify the portions of the image that have been altered. Image tampering and manipulation is typically detected with two kinds of techniques. First, active techniques which involve embedding a watermark to an image when it is taken and authenticating the image by checking the watermark. These techniques have not gained much momentum as they require sophisticated hardware Birajdar and Mankar [1]. Second, passive techniques that asses the integrity of the image based on the content and structure of the data representing the image.

The image processing parts of Java are buried within the java. Awt .image package. The package consists of three interfaces and eleven classes, two of which are abstract. They are as follows:

• The Image Observer inter face provides the single method necessary to support the asynchronous loading of images. The interface implementers watch the production of an image and can react when certain conditions arise. We briefly touched on Image Observer when we discussed the Component class (in Chapter 5, Components), because Component implements the interface.

• The Image Consumer and Image Producer inter faces provide the means for low level image creation. The Image Producer provides the source of the pixel data that is used by the Image Consumer to create an Image.

• The Pixel Grabber and Image Filter classes, along with the Area Averaging Scale Filter, Crop Image Filter, RGB Image Filter, and Replicate Scale Filter subclasses, provide the tools for working with images. Pixel Grabber consumes pixels from an Image into an array. The Image Filter classes modify an existing image to produce another Image instance. Crop Image Filter makes smaller images; RGB Image Filter alters pixel colors, while Area Averaging Scale Filter and Replicate Scale Filter scale images up and down using different algorithms. All of these classes implement

Image Consumer because they take pixel data as input.

2. LITERATURE REVIEW

CASIA Image Tampering Detection Evaluation Database is a widely used standard dataset for evaluating forgery detection. It consists of uncompressed images with various resolution as well as JPEG images with different compression quality factors. The images involve splicing (with arbitrary contour) and also post-processing (blurring and filtering). However, this dataset does not provide the ground truth masks for localization of the tampering operations. Furthermore, the authors argue that there exist statistical artifacts in the way the dataset is built, and might produce unfair results for many forgery detection algorithms. MICC Image Databases is a dataset which aimed at copy move forgery detection and localization. The databases can be further divided into there datasets: F2000, F600, F220, which all contains high resolution images. In each of these datasets, around half of the images are tampered. Only the F600 provides ground truth masks for the tampered images. The type of processing on the copy-move forgeries is limited to rotation and scaling. Dresden Image Database is constructed with the aim of evaluating and developing methods for detecting image tampering as well as identifying the type of device for the acquisition of an image. It contains images taken using 73 digital cameras in 25 different models. They use various camera settings when the authors take the pictures. Columbia Uncompressed Image Splicing Detection Evaluation Dataset [26] provides tampered and original images with image splicing without various post processing techniques applied. It also provides edge ground truth masks for evaluation of the localization of the tampered images. However, the resolution is low and the size of the set is small (e.g., 363 images with 180 tampered and 183 authentic. RAISE Raw Image Dataset consists of 8156 high resolution uncompressed images. The images contain various categories, including outdoor images, indoor images, Landscape and Nature scenes along with People, Objects and Buildings. They have also provided smaller subsets, RAISE-1k, RAISE-2k,
RAISE-4k and RAISE-6k. Uncompressed Colour Image Database [28] was originally a benchmark dataset for image retrieval with the goal of understanding the effects of compression on content based image retrieval (CBIR).

3. PASSIVE AND BLIND LOCALIZATION TECHNIQUES

Passive detection techniques normally do not involve the study of the contents of the image and only concentrate on various image statistics that can be used to discern from non-tampered regions from tampered regions. Some of the techniques involve exploiting the artifacts and inconsistencies that are created due to JPEG compression used widely as an Image format. Some techniques exploit the inherent noise present in the image due to difference in Color Filter array interpolation in different cameras or inconsistencies in the local noise pattern caused due to splicing. Yet another class of algorithm looks at the lighting inconsistency, but as these are primarily semi-automatic techniques, we will not be

evaluating them. The biggest defining characteristics of the algorithms we review is that they do not involve any prior knowledge about the content of the image and only try to detect tampering through statistical means. For a list of algorithms that work only for copy-move operations, [4] provides a survey and benchmark. In general, a copy-move tampering does not need sophisticated techniques. We thus focus more heavily on splicing. For the names of the algorithms, we will use abbreviations according to Zampoglou et al. [21] for convenience.

4. IMAGE PRODUCER AND COLOR MODEL

4.1 Image Producer

The ImageProducer inter face defines the methods that ImageProducer objects must implement. Image producers serve as sources for pixel data; they may compute the data themselves or interpret data from some external source, like a GIF file. No matter how it generates the data, an image producer's job is to hand that data to an image consumer, which usually renders the data on the screen. The methods in the ImageProducer inter face let ImageConsumer objects register their interest in an image. The business end of an ImageProducer—that is, the methods it uses to deliver pixel data to an image consumer—are defined by the ImageConsumer inter face. Therefore, we can summarize the way an image producer works as follows: • It waits for image consumers to register their interest in an image. • As image consumers register, it stores them in a Hashtable, Vector, or some other collection mechanism. • As image data becomes available, it loops through all the registered consumers and calls their methods to transfer the data. There's a sense in which you have to take this process on faith; image consumers are usually well hidden. If you call createImage(), an image consumer will eventually show up. Ever y Image has an ImageProducer associated with it; to acquire a reference to the producer, use the getSource() method of Image. Because an ImageProducer must call methods in the ImageConsumer inter face, we won't show an example of a full-fledged producer until we have discussed ImageConsumer.

4.2 Color Model

A color model determines how colors are represented within AWT. Color Model is an abstract class that you can subclass to specify your own representation for colors. AWT provides two concrete subclasses of Color Model that you can use to build your own color model; they are Direct Color Model and Index Color Model. These two correspond to the two ways computers represent colors internally. Most modern computer systems use 24 bits to represent each pixel. These 24 bits contain 8 bits for each primary color (red, green, blue); each set of 8 bits represents the intensity of that color for the particular pixel. This arrangement yields the familiar "16 million colors" that you see in advertisements. It corresponds closely to Java's direct color model. However, 24 bits per pixel, with something like a million pixels on the screen, adds up to a lot of memory. In the dark ages, memory was expensive, and devoting this much memory to a screen buffer cost too much. Therefore, designers used fewer bits — possibly as few as three,

but more often eight—for each pixel. Instead of representing the colors directly in these bits, the bits were an index into a color map. Graphics programs would load the color map with the colors they were interested in and then represent each pixel by using the index of the appropriate color in the map. For example, the value 1 might represent fuschia; the value 2 might represent puce. Full information about how to display each color (the red, green, and blue components that make up fuschia or puce) is contained only in the color map. This arrangement corresponds closely to Java's indexed color model. Because Java is platform-independent, you don't need to worry about how your computer or the user's computer represents colors. Your programs can use an indexed or direct color map as appropriate. Java will do the best it can to render the colors you request. Of course, if you use 5,000 colors on a computer that can only display 256, Java is going to have to make compromises. It will decide which colors to put in the color map and which colors are close enough to the colors in the color map, but that's done behind your back. Java's default color model uses 8 bits per pixel for red, green, and blue, along with another 8 bits for alpha (transparency) level. However, as I said earlier, you can create your own Color Model if you want to work in some other scheme. For example, you could create a grayscale color model for black and white pictures, or an HSB (hue, saturation, brightness) color model if you are more comfortable working with this system. Your color model's job will be to take a pixel value in your representation and translate that value into the corresponding alpha, red, green, and blue values. If you are working with a grayscale image, your image producer could deliver grayscale values to the image consumer, plus a Color Model that tells the consumer how to render these gray values in terms of ARGB components.

5. BLUNDER LEVEL ANALYSIS

Error level analysis is another method proposed by Krawets. It works by intentionally resaving the JPEG image at a known error rate and then computing the difference between the images. Any modification to the picture will alter the image such that stable areas become unstable. Differently compressed versions of the image are compared with the possibly tampered one by Farid. When the same quality factor of the tampered area is adopted, a spatial local minima, christened as JPEG ghosts by the author, appear and can be used to discern tampered regions. Wang et al. extend the analysis by extracting the high frequency noise from this noise map using Principal Component Analysis and then characterizing the tampered region based on the high frequency noise.

5.1 Block Artifact Grids

For manipulated images, when the tampered part is pasted into the background image, the DCT blocks do not match
and some block artifacts will be left. Li et al. describe a method that uses second order difference of pixel values to extract the Block Artifact Grids and then automatically identify the regions which are likely to be tampered. Ye et al. proposes two methods. The first method uses DCT coefficients to estimate the block artifacts. The second method is by first estimating the DCT

quantization table and then checking the uniformity of the quantization remainders.

5.2. Camera and Local Noise Residuals

5.1.1 Color Filter Array

Image features like Local Noise or Camera Noise arising from the image acquisition process or due to the manufacturing or hardware characteristics of a digital cameras, provide sufficient information to determine an image's authenticity since they are sensitive to image manipulation as well as being difficult to forge synthetically. The methods described in this section are based on the intuition that image regions of different origins may have different noise characteristics introduced by the sensors or post-processing steps of their original source.During acquisition, every pixel receives only a single color-channel value (red, green or blue). To produce the final image, the raw data undergoes an interpolation process, using Color Filter Array (CFA) to obtain a color image with different cameras using slightly different parameters to perform the interpolation. Dirik and Memon, Ferrara et al. exploit the artifacts created by Color Filter Array processing in most digital cameras. Both of the techniques involve estimating CFA interpolation pattern and CFA based noise analysis as features and training a classifier based on these features. This line of attack is more robust than the algorithms mentioned above as they can be applied to images other than those saved in JPEG format. A limitation is that the CFA estimation are sensitive to strong JPEG recompression and resizing.

5.3 Genetic algorithm

Genetic Algorithms (GAs) are procedures that follow the principles of natural selection and natural genetics code, that have proved to be very efficient searching for approximations to global optima in large and complex spaces in relatively short time. The basic components of GAs are:

o genetic operators (mating and mutation)

o an appropriate representation of the problem that is to be solved

o a fitness function

o an initialization procedure

With these basic components of GA, the procedure as follows. It starts with the initialization procedure to generate the first population. The members of the population are basically the strings of symbols (chromosomes) that represent possible solutions to the problem to be solved. Each members of the population for the given generation is evaluated, and, according to its fitness, it is assigned a probability to be selected for reproduction. By using this probability distribution, the genetic operators select some of the individuals. New individuals are obtained, by applying these operators. The mating operator selects two population members and combines their respective chromosomes to create offspring. The mutation operator selects a member of the population and changes some part of its chromosomes. The elements of the populations with the worst fitness measure are replaced by the new individuals.

CONCLUSION

In this paper, we summarized state-of-the-art algorithms for image forgery detection. We

prepared a dataset for benchmarking algorithms. For benchmarking, it may appear anomalous that the detection results are better than the classification ones. However, detection curves are computed only for positive images, i.e. for those images where manipulations exist. It is worth noting that our two benchmarks for detection have different meanings. The one according to IOU is based on the number of correctly and incorrectly detected images, while the one according to the area is based on the number of correctly and incorrectly detected pixels. The latter one has much worse result since the requirement is more strict.

REFERENCES

[1] Ruey-Feng Chang,Wen-Tsuen Chen,and Jia-Shung Wang,"A Fast Finite-State Algorithm for Vector Quantization Design", IEEE Transaction on Signal Processing, Vol.40, No.1, January 1992.

[2] Hong Wong, Ling LU, DA-Shun Que, Xun Luo,"Image Compression Based on Wavelet Transform and Vector Quantization" IEEE proceedings of the First international Conference on Mchine Learning and Cybernetics, Beijing, November 2002.

[3] Y.W. Chen, Vector Quantization by principal component analysis, M.S. Thesis, National Tsing Hua University, June, 1998.

[4] M.F. Barnsley and L.P. Hurd, Fractal Image Compression, AK Peters, Ltd. Wellesley, Massachusetts, 1993.

[5] A.E. Jacquin, Imagecoding basedon a fractaltheory of iterated contractive image transformations. IEEE Trans. on Image Processing, vol. 1, 18-30, 1992.

[6] H.S. Chu, A very fast fractal compression algorithm, M.S. Thesis, National Tsing Hua University, June, 1997.

[7] Y. Fisher, Editor, Fractal Image Compression: Theory and Applications, Springer-Verlag, 1994.

[8] Nitesh Kumar More,Sipi Dubey," JPEG Picture Compression Using Discrete Cosine Transform" International Journal of Science and Research (IJSR), India Online ISSN: 2319-7064

[9] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, Image coding using wavelet transform, IEEE Trans. on Image Processing, vol. 1, 205-220, 1992.

[10] A.S. Lewis and K. Knowles, Image compression using 2D wavelet transform, IEEE Trans. on Image Processing, vol. 1, 244-250, 1992.

Gibson, D., Spann, M., Woolley, S.: A wavelet-based region of
interest encoder for the compression of angiogram video
sequences. IEEE Trans. Inf Technol. Biomed. 8(2), 103–113
(2004)
12. Pearlman, W.A., Islam, A., Nagaraj, N., Said, A.: Efficient lowcomplexity
image coding with a set-partitioning embedded block
code. IEEE Trans. Circuits Syst. Video Technol. 14, 1219–1235
(2004)
13. Do, M.N., Vetterli, M.: The contourlet transform: an efficient

directional multiresolution image representation. IEEE Trans.
Image Process. 14(12), 2091–2106 (2005)
14. Eslami, R., Radha, H.: Wavelet-based contourlet packet image
coding. In: 2005 conference on information sciences and systems.
(2005)
15. Eslami, R., Radha, H.: A new family of nonredundant transforms
using hybrid wavelets and directional filter banks. IEEE Trans.
Image Process. 16(4), 1152–1167 (2007)
16. Singh, S., Kumar, V., Verma, H.K.: Adaptive threshold-based
classification in medical image compression for teleradiology.
Comput. Biol. Med. 37, 811–819 (2007)
17. Chikouche, D., Benzid, R., Bentoumi, M.: Application of the
DCT and arithmetic coding to medical image compression. In:
International conference on information and communication
technologies: from theory to applications (ICTTA 2008), pp. 1–5
(2008). doi:10.1109/ICTTA.2008.4530107
18. Sanchez, V., Nasiopoulos, P., Abugharbieh, R.: Efficient lossless
compression of 4-D medical images based on the advanced video
coding scheme. IEEE Trans. Inf. Technol. Biomed. 12(4),
442–446 (2008)
19. Hearaly, B.C., Viprakasit, D., Johnston, W.K. III.: The future of
teleradiology in medicine is here today. Springer (2008)
20. Xiu-wei, T., Feng, Z.X., Fu, D.T.: Wavelet-based contourlet
coding using SPECK algorithm. In: International conference on
signal processing (ICSP 2008), pp. 1203–1206 (2008). doi:10.
1109/ICOSP.2008.4697346
21. Soman, K.P., Ramachandran, K.I., Resmi, N.G.: Insight into
wavelets-from theory to practice. PHI Learning Pvt. Ltd (2010)
22. Sapkal, A.M., Bairagi, V.K.: Telemedicine in India: a review
challenges and role of image compression. J. Med. Imaging
Health Inf. 1(4), 300–306 (2011)
23. Lalitha, Y.S., Latte, M.V.: Image Compression of MRI image
using planar coding. Int. J. Adv. Comput. Sci. Appl. 2(7), 23–33
(2011)
24. Nguyen, B.P., Chui, C.-K., Ong, S.-H., Chang, S.: An efficient
compression scheme for 4-D medical images using hierarchical
vector quantization and motion compensation. Comput. Biol.
Med. 41, 843–856 (2011)
25. Sevak, M.M., Thakkar, F.N., Kher, R.K., Modi, C.K.: CT image
compression using compressive sensing and wavelet transform.
In: international conference on communication systems and network
technologies (CSNT), pp. 138–142 (2012)
26. Zhu, Z., Wahid, K., Babyn, P., Yang, R.: Compressed sensingbased
MRI reconstruction using complex double-density dualtree
DWT. Int. J. Biomed. Imaging 2013, (2013).
doi:10.1155/
2013/907501

27. Prabhu, K.M.M., Sridhar, K., Mischi, M., Bharath, H.N.: 3-D
warped discrete cosine transform for MRI image compression.
Biomed. Signal Process. Control 8, 50–58 (2013)