International Journal of Scientific Engineering and Applied Science (IJSEAS) - Volume-1, Issue-3, June 2015
ISSN: 2395-3470
www.ijseas.com

IMPROVED JPEG FORENSIC SYSTEM BY USING MULTI LEVEL ERROR ANALYSIS

S.Mohanapriya[1], E.Saranya[2]
[1]M.E, CSE, PG Scholar
[2] Asst.Professor Dept. of CSE
Sir Issac Newton College of Engineering and Technology,
Nagapattinam / Anna University, India.

Abstract:

With powerful graphical editors and sophisticated image manipulation techniques, many powerful photo editing tools in the digital image processing make it extremely easy to modify original images in image processing. Today, image forgery is the big concern in digital forensics industry. The focus of proposed work is to develop a forensic system to detect type of forgeries made in a single pixel change or using internal and external CMFD. The system introduce a method called Multi-Level Error Analysis (MLEA) algorithm to yield a qualifier for each image which is then used to rank images with a high probability of image manipulation. In experimental setup, a set of images was created using digital cameras from different brands (Canon, iPhone, Samsung). A third-party performed copy & move image manipulations on randomly selected images in this set. Each developed ranking method is run at an MLEA quality level of 75%, 85%, and 95%. The gathered results show the best rankings for our methods at an MLEA quality level of 95%. By doing so, hope to reduce the amount of work an expert needs to verify the authenticity of the images. Experiment results show that our proposed scheme is not only robust to multiple copy-move forgery, but also to blurring or nosing adding and with low computational complexity.

Keywords: Image Processing, JPEG compression, MLEA, CMFD,

1. Introduction

Digital images play an ever more important role in today's modern society. Digital cameras are widely available to the general public and with the rise of the current generation, camera integrated, smartphones, images can be shot, edited and shared in a matter of seconds. With image shooting and editing capabilities at a person's fingertips, it is sometimes questionable whether or not images are real. In cases where images are purely used for entertainment purposes, this is not considered a very big issue. However, in cases where images become evidence, it is ever more important to be able to verify the authenticity and integrity of these images. Commonly, images are checked for manipulation by an expert that visually inspects them. This is not a problem when only a small set of images need to be confirmed as usable evidence. However, doing the same for a large sets of images quickly becomes a tedious and time consuming process. The System focuses on the ranking of large sets of images based on the likelihood of being manipulated, as well as the effectiveness and reliability of using such a ranking method. Many different image manipulation techniques exist today. For instance, a person might want to remove the red- eye effect [1] in an image from pupils that have a red glow. Or enhance the image's brightness to improve the appearance of the subject. The most common forms of image manipulation are the so called `copy & move' manipulations [2], [3], and usually entail one of the following techniques: Removing an object from an image, changing the appearance of an object in an image, adding a foreign object to an image. For each of these manipulation techniques, a distinction is made between so called internal and external copy & move image manipulation. With internal copy & move manipulations, an object is copied and moved within the image that is being manipulated. In the other case, with external copy & move manipulations, an object is copied from another image (external source) and moved into the image that is being manipulated. Figure 1.1 and 1.2 illustrate an example of where an object is removed from the original image and is no longer present in the manipulated image. In Figure 1.1, the original image is illustrated where Stalin's commissar of water transport, named Nikolai Yezhov, is clearly present. However, in the manipulated image illustrated in Figure 1.2, Nikolai has been removed completely without leaving any visible traces (to the untrained eye). In this case, an internal copy & move manipulation.

Figure 1.1: Original image

Figure 1.2: Stalin without Yezhov

Secondly Figure 1.3 illustrates an original image of two students of the University of Amsterdam. The image in Figure 1.4 was manipulated and illustrates the addition of a foreign object, namely a white-colored mobile phone that was not part of the original image. Since the mobile phone did not come from the original image itself, an external copy & move manipulation was performed.

Figure 1.3: original image

Figur :1.4 Added Mobile Phone

Not only are there techniques for manipulating images, fortunately there also are techniques for detecting manipulation. Images are stored in different image formats such as Joint Photographic Experts Group (JPEG), Graphics Interchange Format (GIF) [6], Portable Network Graphics (PNG) [7], Tagged Image File Format (TIFF) [8], and RAW [9]. Besides the data that is part of the visible content of an image, these image formats also contain all sorts of additional data called metadata. A popular metadata format used for images is the Exchangeable Image File Format (EXIF) [10]. In most cases the image's associated metadata can be as useful as the image itself. The metadata of an image contains essential information such as its dimensions, but usually also other nonobligatory information such as the brand and model of the camera that took the image, the date and time when being shot, and the color space that was used. Some image manipulation software changes, for instance, the information about the camera model to the name of the program [11]. Looking for these kind of discrepancies is called image format analysis [12] and is considered an active approach of detecting image manipulation [13]. However, image format analysis does not evaluate the image itself but only its metadata. Techniques such as Luminance Gradient (LG), Principal Component Analysis (PCA), Wavelet Transformations (WT), and all explained, allow for the identification of specific image manipulations [12]. These techniques are considered being a passive approach of image manipulation detection and require no prior information about the investigated image or its source [13]. In the case of the mentioned techniques, the results are visualized into another image that contains technique specific highlighting. The highlighting in such a visualization is used by an expert to analyze the questionable image on its authenticity and integrity.

2. Related Work

As said, there exist different image manipulation detection techniques. Although these techniques are all related in the sense that they share a common purpose, namely aiding in the detection of image manipulation, there is a completely different theory underlying each of them. For each of these techniques, in the order LG, PCA and WT respectively, some key aspects will be explained.

2.1 Luminance Gradient

LG is used to identify manipulation of an image by illustrating the general direction of light. To do so, LG utilizes the fact that light rarely hits an object with a uniform intensity. Instead, sections of an object that are closer to the light source will appear brighter. There are many variations of LG, however, they all aim to identify the light direction. In the simplest variation, the image is divided into squared blocks of a fixed size, e.g. 3 by 3 pixels. For every block the direction of light is identified based on adjacent pixels in the block that appear brighter in a certain direction. This is done for every block which results in a set of directions, one direction per block pointing towards the brightest local light source. Finally, the color of every block is remapped based on the direction of the local light source. For example, a direction pointing to the right means all green, to the left means no green, upwards means all red, and downwards means no red. This results in another image that is then used by an expert to determine if the image was manipulated based on the transition of different colored surfaces. Smooth surfaces with even gradient transitions or no transition at all suggest digital manipulation or computer graphics [12].

2.2 Principal Component Analysis

PCA is used to identify the color spectrum within an image. PCA finds patterns in the image data and expresses this pattern data in a certain way to highlight their similarities and differences [14]. Assume an entire image is plotted in three dimensions based on the colors of the pixels: red is mapped to the X-axis, green is mapped to the Y-axis, and blue is mapped to the Z-axis. The resulting plot for most images has a narrow range of colors that appear as a large cluster. Since the image's plot is three-dimensional, there are three Principal Components (PCs). Each PC defines a plane across the plot and emphasizes different sections of information. PC1 identifies the widest variance across the color set, PC2 the second widest variance with respect to PC1, and PC3 the smallest variance. For example, when areas of two different images and color sets are spliced together, they usually end up forming two distinct clusters. With PCA, areas within the image that come from different clusters will have noticeably different values. In the end, PCA is used to detect image manipulation by rendering the distance from each pixel to a PC in another image. Each different PC that is used for rendering will yield a different image that is further analyzed by an expert [12].

2.3 Wavelet Transformations

WT utilizes the properties of wavelets to identify image manipulation. A wavelet is a specific function which is used to analyze signals. Any signal can be decomposed into a set of wavelets that, when combined, give an approximation of the signal. Approximating a signal with wavelets is actually also a form of compression. With images, the image is the signal and all wavelets together approximate this image. To be able to perfectly recreate the image, the total amount of wavelets that are required is equal to the amount of pixels in the image per color channel. Even if only a small percentage of the wavelets are used to render the image, the objects in the image are recognizable even though they are blurry. Using a higher percentage of wavelets causes the image to sharpen up and coloring to increase. Ultimately the entire image should sharpen at the same rate. WT uses this property to aid image manipulation detection. If areas of the image are manipulated by scaling or merging, i.e. using different layers in an image editor, then these areas will sharpen at different rates [12].

3. Proposed System

The proposed system Multi-Level Error Analysis (MLEA) is used to identify image manipulation by detecting areas in an image that have a different level of compression error compared to a given level. Essential is that MLEA makes use of the properties of image formats that utilize lossy compression. Just like the other image manipulation detection techniques explained earlier, applying MLEA to an image results in a visual output in the form of a new image with dimensions equal to the processed image. In this newly generated image, manipulated areas with a different level of compression error stand out because they are visually contrasting in comparison to unmodified areas. An expert analyzes this image to determine if the processed image is authentic.

3.1 Image Representation

Most information in this section comes from [16] which gives an easy to understand introduction to image representation. Images are composed of a large amount of discrete units known as pixels. A pixel essentially is nothing more than a single square, a rectangular region to be precise, set to one specific numerical color value. The amount of horizontal and vertical pixels in an

image determine respectively its horizontal and vertical dimension, better known as resolution. The more pixels an image is composed of, thus the higher the resolution, the smoother shapes in the image become since the density of the pixels increases. To specify a location for each individual pixel, a graphics format is needed. The bitmap graphics format, also known as raster graphics format, is used by the JPEG image format to map a pixel to a certain location in an image.

A color model is required to translate between the color values of pixels and the actual colors that correspond to those values. The possible colors that can be represented by a color model is defined by a so called color space, and determined by the sample precision in bits (1 bit allows for black-and-white, 2 bits allows for 4 colors, et cetera). The most commonly used color model in computer applications is known as RGB, which is short for Red, Green, and Blue. JPEG images however are almost always stored using another color model known as YCbCr. The intensity of an image, referred to as luminance, is expressed by the letter Y. The letters Cb and Cr are referred to as chrominance and specify the blueness and redness of an image, respectively.

Unlike the RGB color model, where all three components are roughly equal, the YCbCr color model concentrates the most important information in the Y component. Doing so allows for an increase in compression of JPEG images by including more data from the Y component than from the Cb and Cr components. As mentioned above, JPEG uses the bitmap graphic format. The main drawback of this type of graphic format is that the storage space that is required to store the image, rapidly grows as the image resolution and color space increases. As explained the solution is to apply compression to the data making up the image.

3.2 Image Compression

At the bit-level, an image can be seen as a set of data. The general idea behind compression is to mathematically reduce the amount of information needed to represent an identical or similar set of data that makes up an image. This is done by exploiting certain patterns in a set of data. When compression is applied on such a set, the storing and transferring of this set is more efficient in comparison to an uncompressed set. Performing compression requires computing power to do the actual work of reducing the stored information that is needed. In turn, compressed data must be decompressed first in order for it to be useful again. There are two types of compression algorithms: lossless and lossy.

3.3 Lossless Compression

Algorithms for lossless compression reduce the total amount of information needed to store an image without losing any of the original image quality. In other words, the original value of every single bit before compression is preserved after decompression. The downside is that lossless compression does not reduce the file size as much as lossy compression would. Lossless compression algorithms are often used when image quality is more important than file size.

3.4 Lossy Compression

Lossy compression algorithms make use of the limitations of the human eye, such as having a hard time to distinguish between nearly identical colors [16]. Some information can be discarded without losing much of the original visual structure. The compression levels in most lossy compression algorithms can be adjusted and as these increase, the file size is reduced, sacrificing image quality due to image degradation. At the highest compression levels, image deterioration becomes more prominent, causing compression artifacts [17].

3.5 JPEG Image Format

Image files in the JPEG image format carry either the 'JPG' or the 'JPEG' file extension. JPEG is an image format that utilizes a compression algorithm. In all cases described in this research report, JPEG is used as a lossy image format. A detailed description of how the JPEG compression algorithm works is beyond the scope of this document. However, in short, the compression algorithm essentially consists of the five steps listed below. For a more detailed description of JPEG you are advised to read [18].

1. Divide the image in blocks of 8 by 8 pixels: The first step of the compression algorithm is to divide the entire image into blocks of 8 by 8 pixels. Each block is then further processed without any relation to the other blocks.

2. Transform the RGB color space to the YCbCr color space: Each pixel within a block is represented by RGB values and is called an RGB vector. The values in an RGB vector usually have significant amount of correlation and therefore need to be converted to something that has less correlation. This is done for better compression results.

3. Apply Discrete Cosine Transformation (DCT): The heart of the JPEG compression algorithm is the Discrete Cosine Transformation. Basically, DCT transforms each block into a so called coefficient which can later be used in the process to decompress the image. DCT relies on the premise that pixels in an image exhibit a certain level of correlation with their

neighboring pixels. Consequently, these correlations can be exploited to predict the value of a pixel from its respective neighbors [19].

4. Apply quantization: The coefficients from the DCT process are stored as integers. Since integers are natural numbers, the coefficients need to be rounded before they can be saved. This is where the quantization comes in. The quantization process is the actual part of the compression algorithm that makes it a lossy compression algorithm since rounding the coefficients to integers will lead to losing some of the original data.

5. Apply Huffman encoding: The last step in the compression algorithm is to encode the transformed and quantized image. For this purpose, the Huffman encoding technique is used. The idea behind Huffman coding is to identify pixels that occur frequently in an image and assign them short bit representations. Pixels that occur infrequently in an image are assigned long bit representations.

3.6 Error Analysis

As mentioned, JPEG utilizes lossy compression. This means that if an image is saved using the JPEG image format, some information is lost due to quantization. The losing of information introduces what is known as error. The amount of information that is lost and the amount of error that is gained, is determined by the level of quality the JPEG image is saved at. JPEG quality levels are expressed in percent by values ranging from 0 up to 100. The basic rule here is that the higher the numerical value of the quality level, the less information is lost. Furthermore, resaving a JPEG image will impact the quality of the image, even though no changes were made. For example, if an image initially compressed at a quality of 90% is resaved at a quality of 90%, the result is equivalent to a one-time save at 81%. This is calculated by taking 90% of 90%, essentially meaning the nth resave at 90% should be approximately equivalent to 90%n [12], [20]. Another example is where an image at an initial quality of 75% is resaved at a quality of 90%, then the resulting image will have a quality of 67.5%. It is interesting to note that the amount of information that is lost by each resave is not linear. The amount of error introduced by each save, is limited to the 8 by 8 blocks used by the compression algorithm of JPEG. However, when an image is (partially) modified, the 8 by 8 blocks containing modification are no longer at the same level of quality as the rest of the unmodified blocks. The way MLEA works is by resaving a potentially manipulated image at a certain given quality level, e.g. 95%. Doing so intentionally introduces a known error rate. Each 8 by 8 block at the same location in both images, namely the potentially manipulated image and the resaved image, are compared and the error state difference between the two is calculated. According to [12], if there is little to no difference, this indicates that the block in the resaved image has reached its minimal error state at the given quality level (in this example 95%). However, if there is a substantial amount of difference, then the block is not at its minimal error state. This information is translated into another image, which has said has a resolution identical to the potentially manipulated image, where for each block the amount of difference in error is visibly expressed by different levels of brightness. The closer the given quality level is to that of a block, the smaller the difference will get and the darker this block will appear in the output image. To illustrate this, a few examples follow.Figure1.5 illustrates an original image at a quality of 96%. Figures 1.6, 1.7 and 1.8 illustrate the MLEA output images at a quality of 75%, 85%, and 95%, respectively. MLEA indicates that the difference in error for all blocks is very small.

Figure:1.5 Original image

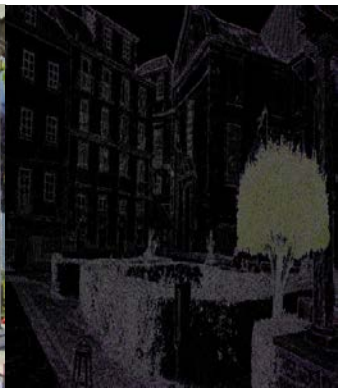

Figure: 1.6 MLEA at 75%

Figure: 1.7 MLEA at 85%

Figure:1.8 MLEA at 95%

4. Conclusion

The MLEA algorithm can be used to detect image manipulation techniques, an expert would still need to manually verify each

ranked image to determine if it is manipulated. Not enough certainty can be given that manipulated images get ranked higher than authentic images, and therefore the ranking will not help in aiding an expert by reducing the amount of work. To conclude, MLEA can be used to rank a set of image. By using our methods, the set of images can be sorted with some of the manipulated images. However, when all manipulated images need to be found in a set, an exhaustive search by inspecting each image manually still seems the only viable option.

5. References

[1] S. Preston, \The science behind the red-eye e_ect," March 2011. http://www.cameratechnica .com/2011/03/14/ the-science-behind-the-red-eye-effect/.

[2] M. Zimba and S. Xingming, \Dwt-pca (evd) based copy-move image forgery detection," tech.rep., The School of Computer and Communications, Hunan University, January 2011.

[3] T. Shahid and A. B. Mansoor, \Copy-move forgery detection algorithm for digital images and a new accuracy metric," tech. rep., College of Aeronautical Engineering, National University of Sciences and Technology, November 2009.

[4]\The commissar vanishes," September 1999. http://www.newseum.org/berlinwall/ commissar_vanishes/vanishes.htm.

[5]D.Lucas,\Katie couric's weight loss," 2006. http://www.famouspictures.org/mag/index.php?title=Altered_Images#Katie_Couric.27s_weight_loss_-_2006.

[6] CompuServe, \Graphics interchange format(sm)," tech. rep., PNG Development Group, 1999.

[7] M. Adler, T. Boutell, and J. Bowler, \Png (portable network graphics) speci_cation, version 1.0," tech. rep., CompuServe Incorporated, 1990.

[8] A. D. Association, \Ti_ revision 6.0," tech. rep., Adobe, June 1992.

[9] M. Reichmann, \Understanding raw _les explained." http://www.luminous-landscape.com/tutorials/understanding-series/u-raw-files.shtml.

[10] \Exchangeable image _le format for digital still cameras: Exif version 2.2," tech. rep., Japan Electronics and Information Technology Industries Association, April 2002.

[11] R. Tortorella, \Image doctoring: Jpeg encoding and analysis," tech. rep., National Aviation Reporting Center on Anomalous Phenomena, May 2009.

[12] N. Krawetz, \A picture's worth, digital image analysis," tech. rep., 2007.

[13] \Detecting image forgery - state of the art," March 2009.

[14] L. I. Smith, \A tutorial on principal components analysis," tech. rep., University of Otago, February 2002.

[15] D. Santa-Cruz and T. Ebrahimi, \An analytical study of jpeg 2000 functionalities," tech. rep., Swiss Federal Institute of Technology, September 2000.

[16] J. Miano, Compressed Image File Formats JPEG, PNG, GIF, XBM, BMP. ACM Press, July 2009.

[17] M.-Y. Shen and C.-C. J. Kuo, \Review of postprocessing techniques for compression artifact removal," tech. rep., March 1998.

[18] D. Austin, \Image compression: Seeing what's not there." http://www.ams.org/samplings/feature-column/fcarc-image-compression.

[19] S. A. Khayam, \The discrete cosine transform (dct): Theory and application," tech. rep., Department of Electrical & Computer Engineering, Michigan State University, March 2003.

[20] N.Krawetz, \Resaving images," February 2010. http://www.hackerfactor.com/blog/index.php?/archives/354-Resaving-Images.html.