A Novel Method for Detecting Double Compressed Facebook JPEG Images

Allan NG, Lei Pan, and Yang Xiang

School of IT, Deakin University, Melbourne, Australia
{ngall,l.pan,yang}@deakin.edu.au

Abstract. Images published on online social sites such as Facebook are increasingly prone to be misused for malicious purposes. However, existing image forensic research assumes that the investigator can confiscate every piece of evidence and hence overlooks the fact that the original image is difficult to obtain. Because Facebook applies a Discrete Cosine Transform (DCT)-based compression on uploaded images, we are able to detect the modified images which are re-uploaded to Facebook. Specifically, we propose a novel method to effectively detect the presence of double compression via the spatial domain of the image: We select small image patches from a given image, define a distance metric to measure the differences between compressed images, and propose an algorithm to infer whether the given image is double compressed without referring to the original image. To demonstrate the correctness of our algorithm, we correctly predict the number of compressions being applied to a Facebook image.

1 Introduction

The general public has no easy method to check the authenticity of a Facebook image. Furthermore, image forensics fails to address the issues of a booming number of retouched images on Facebook. According to Sencar and Memon [11], image forgery detection is "the process to determine whether an image has been manipulated or processed after it was captured by an acquisition device like a digital camera". This definition fails on Facebook images because each uploaded image is compressed by Facebook image filter before publication. Moreover, the traditional viewpoint [9] that the original image and/or the device used to shoot the image should be confiscated prior to the investigation becomes rarely applicable. Hence, we need to develop a new forensic paradigm whilst handling Facebook images.

Specifically, JPEG images are difficult for digital forensic investigators because JPEG is a lossy compression algorithm which prevents the original raw image from being 100% restored from the JPEG image. Generally, there are three steps when a JPEG image is generated — firstly, a raw image is partitioned into 8 × 8 blocks; secondly, a two dimensional Discrete Cosine Transform (DCT) is applied to each block; thirdly, the DCT coefficients are converted to integers by using JPEG quantization table which is specific to each camera device or each

L. Batten et al. (Eds.): ATIS 2014, CCIS 490, pp. 191–198, 2014.

image editing software tool. Hence, the retouched Facebook JPEG images are highly likely to contain digital traces left by the application of multiple DCT operations. Double compression is referred to as a process when a JPEG image file is decompressed and altered to meet an editing tool's specification before the modifications are saved to the image [12]. Detection of double compression helps to determine whether the image has been amended, according to [2] and [10].

In this paper, we solve the problem of detecting double compression in Facebook JPEG images without using its original image. Our assumptions are threefold — 1) digital forensic investigators cannot easily obtain the original photo or original device which is used to shoot the photo; 2) we can obtain or infer the camera device information to obtain the quantization table for the camera; 3) Facebook image filter invariantly compresses the uploaded JPEG images though its parameters are not disclosed to the public.

We define a novel metric to measure the distance between two JPEG images in which one image is obtained by compressing the other. Based on this metric, we could determine how many rounds of compression exist between two images. Furthermore, by using the color intensity information of the selected 8×8 patches, we construct a reference image which reaches the limit after a number of compression rounds. Hence, we estimate the presence of double compression by measuring how many compression rounds are there in the given JPEG image before reaching the maximum. We also conduct a real life experiment to demonstrate the effectiveness of our method.

The rest of this paper is organized as follows: Section 2 surveys the related work and identifies the research gap. Section 3 presents our novel distance metric for images and our double compression detection algorithm. Section 4 is a case study when we successfully and correctly determine the number of compressions applied to a Facebook JPEG image. Section 5 concludes the paper and discusses future work.

2 Related Work

Image forgery detection uncovers manipulated or tampered digital images and attempts to distinguish them from their original counterparts. A common approach to manipulating images is the copy-move method which is employed to tamper with images by altering the content within an image, commonly referring to a particular object within the scene being replaced or substituted with some other form in the same image, thus creating a new forged image. Ardizzone et al. [1] use texture descriptors to detect this type of forgery, which exploits texture as features extracted from blocks, because the block-matching process is an integral part of the copy-move method. Conversely, a process applying blind image forensics generates more robust results to detect the use of copy-move method, where the key idea is to divide the image into smaller blocks for analysis. The main process applies a Discrete Wavelet Transform (DWT) algorithm to divide a compressed image into overlapping blocks each of which has a fixed size; these blocks are firstly sorted by a lexicographic algorithm and then the duplicated blocks are identified using phase correlation [6].

Most research efforts focus on analyzing the DCT coefficient to detect double-compressed images. Fei and Xi-lan [3] measure the global blur of the image and exploits the DCT information within the image; working from sub-block to sub-block, the authors claim that we can ultimately obtain the blur region and locate the false blur regions inside the image. Lukàš and Fridrich [8] identify that the DCT coefficients lose their integer values during the quantization phase, based on which they propose to recover the missing values which are replaced by zeros according to the coefficients. Specifically, these missing values could provide a reliable indicator of identifying the primary quantization matrix used to generate the JPEG image. Moreover, Lukàš and Fridrich [8] suggest that the existence of double peaks in the DCT coefficient histogram is a good indicator of the application of double compression to an JPEG image.

Hou et al. [4] advocate detecting of double compression by extracting the first digit features of DCT coefficients. That is, the first digit should be zero in value if the image has been double-compressed. Furthermore, Popescu [9] attempts to detect double compression by observing any inconsistent patterns in the JPEG coefficient histogram, which is based on the belief that the digital trace of specific correlations of the image is often left in the image after parts from different images are merged into one JPEG file.

Furthermore, Huang et al. [5] achieve accurate results by using a random perturbation strategy to detect double compression. This method detects the double compression based on single compressed JPEG images even if the same quantization matrix is used in the process of compressing JPEG images. Lastly, Liu et al. [7] attempt to identify and re-compress the misaligned cropping pixels which will give a different reading in DCT coefficients because tampering activities such as copy and past are often done to an image patch which almost always has different DCT coefficients to the original image.

In summary, the above techniques rely on strict assumptions — the investigator has to confiscate the camera device; the investigator has prior knowledge of the original image and of the patch; the investigator has to analyze the entire image which requires excessive amount of computation and time. To address these issues, we propose a novel metric on pixel intensity in a small 8×8 patch of JPEG images in Section 3.

3 A New Metric of the Change of Color Intensity

Suppose there are two JPEG images I_i and I_j both of which are generated by compressing an original raw image I_0. Without using the original image, our method measures I_i and I_j to infer which image is compressed more and to estimate the number of the compression. To avoid confusion, we denote I_0 as the original raw image, I_1 as the first compressed JPEG image, I_2 as the double-compressed JPEG image, I_3 as the triple-compressed JPEG image, $\ldots$, and so on. Hence, I_i stands for the image which is compressed for i times.

Definition 1. *Suppose that we have two* 8×8 *JPEG image patches* I_i *and* I_j *both of which are from the same origin, the maximal difference in color intensity is defined as follows*

$$P_{max}^{i,j} = \max\{P_x^i - P_x^j \mid x \in \{0, 1, \ldots, 63\}\},$$

where P_x *is the RGB tuple at the position* x*. Similarly, the minimal difference in color intensity is*

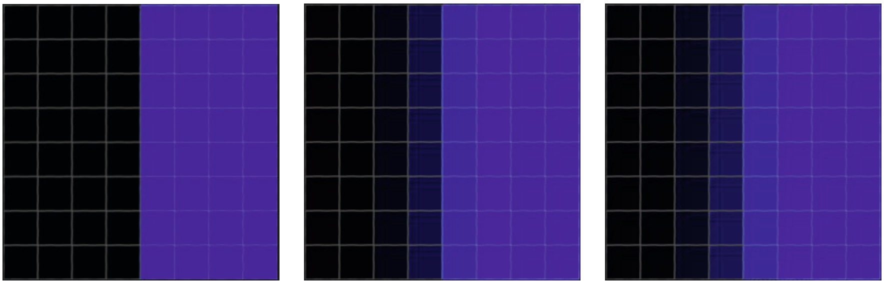
$$P_{min}^{i,j} = \min\{P_y^i - P_y^j \mid y \in \{0, 1, \ldots, 63\}\},$$

where P_y *is the RGB tuple at the position* y*.*

Definition 2. *For two* 8 *patches of* I_i *and* I_j*, we define a new distance metric as*

$$distance(i, j) = P_{max}^{i,j} - P_{min}^{i,j}. \tag{1}$$

Example 1. We use a camera on a Samsung Galaxy S3 mobile to shoot a photo of a color-board consisting of two colors only — black and blue. Figures 1(a), 1(b) and 1(c) are raw image I_0, single compressed JPEG I_1 and doubled compressed JPEG I_2, respectively. Please note that the gradient edge is becoming smoother as more compression is applied.



(a) I_0 (b) I_1 (c) I_2

Fig. 1. Three 8×8 Patches Containing Black and Blue

The red channel intensity values are all zeros for I_0; I_1 has a slight higher reading in the middle of each row so that the values become

$$\{\underbrace{0, 0, 0, 0, 7, 0, 0, 0}_{8 \text{ times}}\};$$

and I_2 has

$$\{\underbrace{4, 0, 0, 0, 13, 3, 2, 1}_{8 \text{ times}}\}.$$

The green channel is similar to the red channel: I_0 has all zeros; I_1 has

$$\{\underbrace{0,0,0,0,8,0,1,1}_{8\text{ times}}\};$$

and I_2 has

$$\{\underbrace{0,0,0,0,11,1,3,2}_{8\text{ times}}\}.$$

The blue channel has different readings: I_0 has

$$\{\underbrace{0,0,0,0,255,255,255,255}_{8\text{ times}}\};$$

I_1 has

$$\{\underbrace{0,0,0,57,197,254,254,254}_{8\text{ times}}\};$$

I_2 has

$$\{\underbrace{0,7,29,79,182,224,244,252}_{8\text{ times}}\}.$$

The maximal difference between I_0 and I_1 occurs on the fifth column of the patches such that $P^{0,1}_{max} = 1$, and the minimal difference between I_0 and I_1 occurs on the fourth column such that $P^{0,1}_{min} = -57$. Hence the distance between I_0 and I_1 is $distance(0,1) = 58$. Similarly, we derive the distance between I_1 and I_2 as $distance(1,2) = 26 - (-29) = 55$ and distance between I_0 and I_2 as $distance(0,2) = 49 - (-79) = 128$.

Our distance metric does not always satisfy the triangular inequality. As observed in Example 1, we have $distance(0,2) > (distance(0,1) + distance(1,2))$. This is simply caused by the max and min functions. Furthermore, the distance values between the compressed images tend to converge to zero when they are more compressed. Hence, we develop the following algorithm to detect double compression of JPEG images without referring to the original picture.

Our algorithm consists of the following five steps:

1. Divide a given image I_i into 8×8 patches and randomly select a handful of the patches each of which is primarily composed of two colors. Usually, we can find these patches in the edge of objects so that approximately 32 pixels have one color and the rest have the other.
2. Compress each patch and calculate the distance values between the compressed ones and the ones obtained in **Step 1**.
3. Repeat **Step 2** until the distances converge and record the number of compression rounds k.
4. Use the same quantization table to generate a reference photo $\widehat{I_0}$ which has the color setting similar to I_i. And then repeat the first three steps for $\widehat{I_0}$. At the end of the process, we obtain the number of compression rounds $\hat{k}$ for $\widehat{I_0}$.
5. Calculate $\widehat{k} - k$. If the result is greater than 2, we conclude the use of double compression on the given image I_i.

4 Case Study

To demonstrate the correctness of our detection method, we apply our algorithm on a Facebook image as shown in Figure 2. The picture is shot by using a Samsung Galaxy S3 mobile phone. We select an 8 × 8 patch from the boy's shoulder area which is primarily blue and black. For privacy reasons, we have blurred the image but highlighted the selected patch.

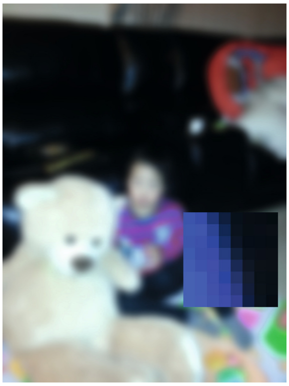


Fig. 2. A Facebook Image with a Selected Patch

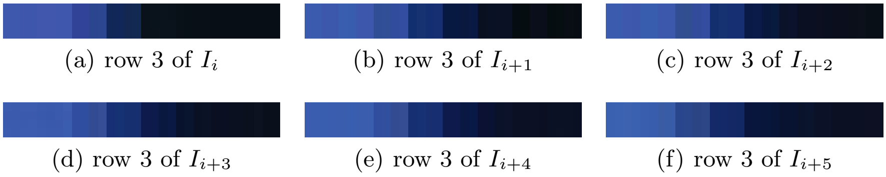
We follow the second step of our algorithm by compressing the selected patch with the Facebook image filter. And we obtain a series of 6 images denoted as I_i, I_{i+1}, . . . and I_{i+5}. We calculate the distances between these images by using Equation 1. The results are listed below.

	I_i	I_{i+1}	I_{i+2}	I_{i+3}	I_{i+4}	I_{i+5}
I_i	0	31	33	34	42	44
I_{i+1}		0	5	11	19	23
I_{i+2}			0	6	8	17
I_{i+3}				0	8	11
I_{i+4}					0	4
I_{i+5}						0

Each image has zero distance to itself; the distance values accumulate when the image is more compressed; the biggest distance is $distance(i, i+5) = 44$, and the smallest non-zero distance is $distance(i + 4, i + 5) = 4$. Furthermore, the above distance values are consistently decreasing when the image is more compressed.

Moreover, the distance value between the fifth and the sixth compressed images is zero. That is, $distance(5, 6) = 0$. Hence, we derive that the number of compression rounds is $k = 5$.

In order to construct a reference image, we observe the pattern of the row which produces the peak readings for both P_{max} and P_{min}. In our case, we identify the third row from the 6 patches as listed in Figure 3. Because blue and black are dominant colors in these patches, we set the reference image as the black and blue image as shown in Figure 1(a). And, we obtain the number of compression rounds for the reference image as $\hat{k} = 7$.



(a) row 3 of I_i

(b) row 3 of I_{i+1}

(c) row 3 of I_{i+2}

(d) row 3 of I_{i+3}

(e) row 3 of I_{i+4}

(f) row 3 of I_{i+5}

Fig. 3. The Third Row of the 6 Image Patches

Now we calculate $\hat{k} - k = 7 - 5 = 2$. Because this result is not greater than 2, we conclude that Figure 2 is not double compressed according to our algorithm. So we make a correct conclusion because this photo is single compressed.

As we observed in this case study, the intersection parts of each patch (as shown in Figure 3) are visually smoother after each compression, which matches the normal behavior of DCT-based compression. Our distance metric is a simple but reliable means to measure the differences between compressed image of the same origin. A potential drawback of our algorithm could be robustness, because malicious attackers might be able to affect our distance metric by modifying the color intensity values of the brightest and darkest pixels in each 8 × 8 patch block.

5 Conclusions and Future Work

This paper introduces a novel, simple and effective distance metric for detecting double compressed Facebook JPEG images. We also propose an algorithm to infer the number of rounds that a given Facebook image has been compressed. Specifically, our algorithm detects whether a Facebook image is modified from another Facebook image without referring to the original image. Our case study demonstrates the correctness of applying our distance metric on a real Facebook photo.

Because we rely on the color intensity, our detection algorithm works well on simple filters used by Facebook. The effectiveness of this method is unknown for sophisticated filters which is capable of sharpening the image after modification. As part of future work, we plan to conduct more testing cases on the complex filters included in professional image editing tools such as *Adobe Photoshop*.

References

1. Ardizzone, E., Bruno, A., Mazzola, G.: Copy-Move Forgery Detection via Texture Description. In: Proceedings of the 2nd ACM Workshop on Multimedia in Forensics, Security and Intelligence (2010)

2. Bianchi, T., Piva, A.: Analysis of Non-Aligned Double JPEG Artifacts for the Localization of Image Forgeries. In: Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–6 (2011)
3. Fei, P., Xi-lan, W.: Digital Image Forgery Forensics by Using Blur Estimation and Abnormal Hue Detection. In: Proceedings of the Symposium on Photonics and Optoelectronic (SOPO), pp. 1–4 (2010)
4. Hou, W., Ji, Z., Jin, X., Li, X.: Double JPEG Compression Detection Base on Extended First Digit Features of DCT Coefficients. International Journal of Information and Education Technology 3(5), 512–515 (2013)
5. Huang, F., Huang, J., Shi, Y.: Detecting Double Compression with the Same Quantization Matrix. IEEE Transactions on Information Forensics and Security 5(4), 848–856 (2010)
6. Khan, S., Kulkarni, A.: Robust Method for Detection of Copy-Move Forgery in Digital Images. In: Proceedings of the International Conference on Signal and Image Processing (ICSIP), pp. 69–73 (2010)
7. Liu, Q., Li, X., Cooper, P., Hu, X.: Shift-Recompression-Based Feature Mining for Detecting Content-Aware Scaled Forgery in JPEG Images. In: Proceedings of the 12th International Workshop on Multimedia Data Mining (MDMKDD), pp. 10–16 (2012)
8. Lukàš, J., Fridrich, J.: Estimation of Primary Quantization Matrix in Double Compressed JPEG Images. In: Proceedings of the Digital Forensic Research Workshop (2003)
9. Popescu, A.: Statistical Tools for Digital Image Forensics. Ph.D. thesis, Department of Computer Science, Dartmouth College, Hanover, PhD thesis (2005)
10. Qu, Z., Luo, W., Huang, J.: Identifying Shifted Double JPEG Compression Artifacts for Non-intrusive Digital Image Forensics. In: Hu, S.-M., Martin, R.R. (eds.) CVM 2012. LNCS, vol. 7633, pp. 1–8. Springer, Heidelberg (2012)
11. Sencar, H., Memon, N.: Overview of State-of-the-art in Digital Image Forensics. Statistical Science and Interdisciplinary Research, pp. 1–19 (2008)
12. Thing, V., Chen, Y., Cheh, C.: An Improved Double Compression Detection Method for JPEG Image Forensics. In: Proceedings of the IEEE International Symposium on Multimedia, pp. 290–297 (2012)