SCC0251 - Image Processing
Prof. Moacir Ponti

Final Project Partial Report - Group 2
Detection of digital alterations in real images

Ariella Yamada Brambila - 8937034
Rodrigo de Andrade Santos Weigert - 8937503
Universidade de São Paulo
São Carlos

11/06/2017

1 Introduction

As we stated in the project description submitted previously, our original objective was to detect specific kinds of digital image tampering: addition, removal and deformation of scene elements. We did quite a bit of research on possible techniques we could use to achieve this goal, and in the following sections we present the findings we considered most relevant. We also discuss our plans for the final project and how our findings affect our goal.

2 Techniques

It turns out that attacking our problem might be harder than we expected. During our research we found a couple of very different techniques, with a wide range of sofistication levels and reliability. Some were more general, and some targetted more specific problems such as the cloning (or copy-move) detection. In the following subsections we talk about some of the methods/studies we think are worth mentioning, as well as our thoughts about them.

2.1 Demosaicing Artifacts

Reference [1] presents two methods for detection of tampering in general ("*Universal Tamper Detection*") based on the demosaicing process of the digital camera. The first method [1, section 2.1] seemed too complex for our knowledge and time constraints, but the second one [1, section 2.2] looked more promising. Here is a very brief explanation of what we were able to comprehend about this method:

- It assumes (reasonably, we suppose) that the images were taken by a camera that uses the Bayer Color Filter Array (CFA) shown in Figure 1 over its pixel sensors.
- It starts by applying an algorithm called dual tree wavelet based denoising [2] to the green channel of the image. We don't know if this can be done easily enough.
- Using the assumption of the first item, it separates into two vectors, A_1 and A_2, the pixels which had green channel intensity directly measured (non-interpolated) and those which had the value calculated by interpolation with the neighboring pixels.
- Finally, it decides whether an image has been tampered with by using a very simple measure based on the variance of A_1 and A_2.

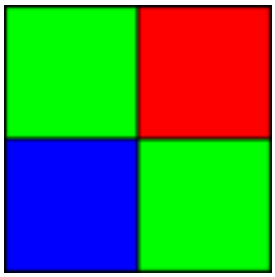

Figure 1: Bayer CFA

We have a relevant question about this method, to which we don't know the answer yet. If the image was cropped, couldn't this cause the CFA alignment to be different from expected (e. g. the top left pixel of the image could be one with a red filter instead of a green one) and thus ruin the algorithm, which relies on the knowledge of where the green filters are?

2.2 Binary Similarity Measures

Reference [3] considers the binary images obtained from different bit planes of the original image and proposes a method for general tampering detection based on Binary Similarity Measures (BSMs) computed between these bit plane images. The used BSMs are based on the 4-neighborhood of the pixels. Several different BSMs were used, some of them calculating correlations within a bit plane, and others between different bit planes. The BSMs were considered as features and the best ones were selected by an algorithm (Sequential Floating Forward Search). Then a linear regression classifier was used.

We are confident that it wouldn't be too difficult for us to implement the computation of the BSMs. We are not so sure about the last steps, though.

2.3 Error Level Analysis and JPEG Compression

Another method that we found analyzes the JPEG compression of the photo to detect which part of it has been tampered with. As we know, JPEG format uses a lossy compression algorithm, so every time that a photo is re-saved, it suffers compressions. The Error Level Analysis [4] is based on re-saving images, and computing the differences between them.

If a photo hasn't been through any altering process, then the compression is applied uniformly to the data set, resulting in an uniform level of compression artifacts. But if, the image has suffered some kind of alteration, the areas that have been altered presents different levels of compression. So when we calculate the difference between the results of compression and the original picture, we can identify unstable areas on it, which represents possible alterations to the image.

This method seems simple to implement, but it hasn't presented good accuracy in the few tests we ran. The paper [5] also says that this method doesn't do much by itself, but it can be combined along with other complimentary analysis.

The JPEG Compression method studied in [5] states that when a photo that has been doctored is compressed, the unchanged region undergoes double JPEG compression, and the doctored region can be consider to be compressed only once. But if the compression is with high quality, the high frequencies were probably erased by previous compressions, so by next compression, noise analysed is only based in medium and low frequencies. However, the tampered region (that problably didn't undergo any previous compression) still contains low, medium and high frquency noise. The results of compression of tampered images presents different noise level regions, this differences can allow us to identify the altered regions. This identifying is done using Discrete Cosine Transform (DCT) and Principal Component Analysis (PCA). We yet don't know if we'd able to reproduce the method without unreasonable effort.

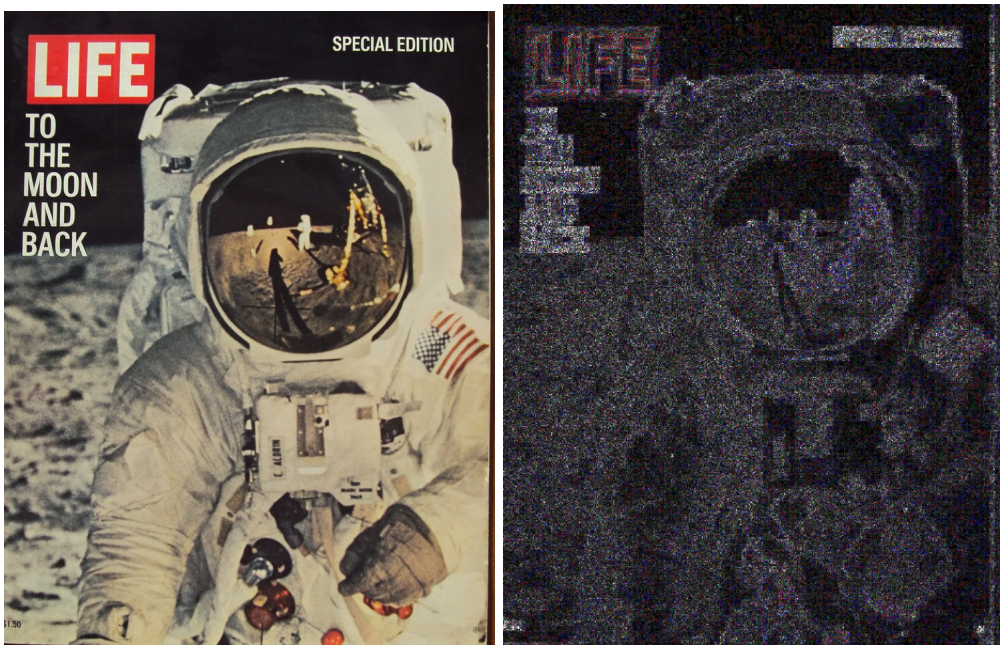

Figure 2: ELA applied to the Moonwalk picture. Presents clear differences between the title of the maganize and some detail on the helmet of the astrounaut with the background and spacesuit.

Figure 3: The famous Alf Kid picture, with alterations on it's t-shirt. ELA applied to the tampered picture.

2.4 Least Significant Bit Patterns?

We were told about a possible technique that looks for patterns in the least significant bits of an image. Unfortunately, we were unable to find any references regarding such method. We implemented a program that takes images and an arbitrary 8-bit mask M as input, (optionally) converts it to grayscale, applies the logical AND operation between each pixel of the images and M, and then normalizes and displays the results. A few, manual tests were ran with this program and little to no visual patterns could be identified. The best one we found is shown in Figure 4.

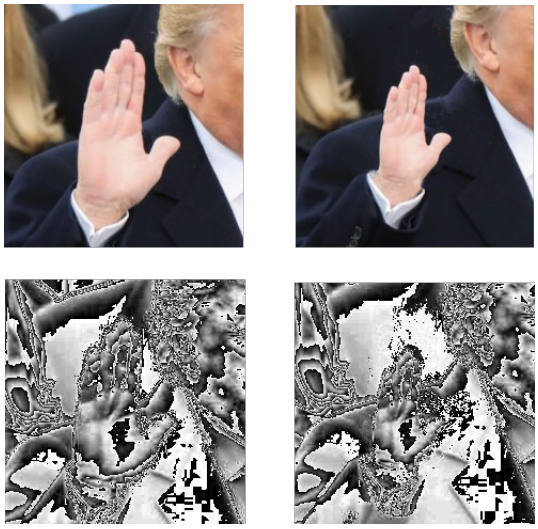

Figure 4: $M = 31_{10} = 00011111_2$. There's a barely distinct white spot in the space the hand no longer occupies.

3 Reaching the Objective

3.1 Steps

Because of the amount of uncertainties, we're still unsure of the steps we're going to take towards the final project. Here are some possibilites we consider:

- Just implement and analyze one of the techniques of the references. ELA is the simplest one, but maybe it's *too* simple.
- Look more thoroughly into the least significant bit patterns and see what we can get out of them.
- Try to fully implement two or more techniques described and compare their accuracy.
- Try and devise a new technique based on the ones in the previous section. Maybe a simplification of one or more of them combined.
- Look for other techniques and try to find one that better fits our purpose and constraints.

Our original objective also will most likely have to be slightly adapted, depending on our next choices. We may have to:

- Focus on a more specific kind of digital image doctoring, e. g. just artifact insertion (clipping) or just reshaping/resizing.
- Give up on trying to locate the tampering in the image, as some methods we described don't have that capability.

3.2 Image Sources

The nature of our project is such that our results are better evaulated by humans than machines (except maybe if we opt to just decide whether an image has been tampered with or not). Because of that, and because we still don't know if a large database of doctored/non-doctored images exists, our idea is to just handpick images from around the internet. Two places where we can do that are `reddit.com/r/picrequests` and `reddit.com/r/photoshopbattles`. We can also select famous pictures that are known to be edited, or even edit our own images.

In order to generate many images from one, we can divide it in doctored and non-doctored blocks. However, we don't think this will be necessary. In our opinion, we shouldn't too many test images. Then again, we're very uncertain of our next steps and things can still change.

4 Final Thoughts

We hope this report showed that we are not sure on what to do next, and why so. We are aware that this uncertainty is not ideal, but at least this report can (hopefully) show that it isn't for lack of trying. We found several methods we could use, but more often than not we're unable to fully understand *how* they work or *why* they work. This makes it harder for us to find an ideal strategy, which would be one that both obtains decent results and isn't too difficult or too trivial to implement. Maybe we can get some advice on what to do next. Maybe we'll severely underperform on this project and fail the course. The possibilites are endless. We'll see.

References

[1] Ahmet Emir Dirik and Nasir Memon. "Image Tamper Detection Based on Demosaicing Artifacts". In: *Proceedings of the 16th IEEE International Conference on Image Processing*. ICIP'09. Cairo, Egypt: IEEE Press, 2009, pp. 1481–1484. ISBN: 978-1-4244-5653-6. URL: http://dl.acm.org/citation.cfm?id=1818719.1819143.

[2] Levent Sendur and Ivan W. Selesnick. "Bivariate Shrinkage With Local Variance Estimation". In: *IEEE SIGNAL PROCESSING LETTERS* 9.12 (Dec. 2002), pp. 438–441.

[3] S. Bayram et al. "Image manipulation detection with Binary Similarity Measures". In: *2005 13th European Signal Processing Conference*. Sept. 2005, pp. 1–4.

[4] Neal Krawetz. *A picture's worth...* USA, 2007.

[5] Wei Wang, Jing Dong, and Tieniu Tan. "Tampered Region Localization of Digital Color Images Based on JPEG Compression Noise". In: *Digital Watermarking: 9th International Workshop, IWDW 2010, Seoul, Korea, October 1-3, 2010, Revised Selected Papers*. Ed. by Hyoung-Joong Kim, Yun Qing Shi, and Mauro Barni. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 120–133. ISBN: 978-3-642-18405-5. DOI: 10.1007/978-3-642-18405-5_10. URL: http://dx.doi.org/10.1007/978-3-642-18405-5_10.