Automatic Tampering Detection in Spliced Images with Different Compression Levels

Diego García-Ordás[1], Laura Fernández-Robles[2], Enrique Alegre[2], María Teresa García-Ordás[2], and Oscar García-Olalla[2]

[1] University of León
diego.ordas@gmail.com
[2] León University
{l.fernandez,enrique.alegre,ogaro,mgaro}@unileon.es

Abstract. In this paper, we introduce a blind tampering detection method based on JPEG ghosts [3] capable of detecting tampering when it is created by splicing regions with different compression levels in an image. Given an image, a set of re-compressions of that image is generated and used to extract a feature vector to train a Support Vector Machine classifier. We used two different datasets in our experiments. The first one, extracted from Columbia Uncompressed Image Splicing Detection, was used to compare results with other works. Our method outperformed the previous ones when dealing with small tampered regions and similar qualities, offering hit rates above 97% (100% in the case of non-tampered images). With the second dataset, CASIA1 Tampered Image Detection Evaluation Dataset, our method offered a hit rate of 98.71% when discerning between the original and the spliced image, with just a 0.44% of non-tampered images wrongly classified.

Keywords: Tampering, JPEG, Compression, spliced images, SVM.

1 Introduction

Modern photo editing software like Adobe Photoshop, GIMP and similar programs allow their users to easily edit digital images, and as a result, image tampering is a very frequent operation in digital photographies. Being able to discern between an image that has suffered any kind of tampering and the original image is very important in several kind of scenarios, for example images presented as evidences at courts or images used for data extraction purposes. There are several kinds of tampering: splicing, compression, scale changes, rotation, etc. In this paper we focus on the detection of a specific kind of tampering, the one created by splicing (copying and pasting some regions of an image 'A' in another image 'B') when 'A' and 'B' have different qualities or compression levels.

Different tampering detection methods are covered in the literature. Some methods, such as [7,13,11], need embedding watermarks on the original images in order to detect tampering. On the contrary, we are interested in those called

J.M. Sanches, L. Micó, and J.S. Cardoso (Eds.): IbPRIA 2013, LNCS 7887, pp. 416–423, 2013.

blind tampering detection. Therefore, we do not have access to the original image and there is no need of previous image protection using neither watermarks nor any other preprocessing technique.

In this way, in [1,2,14], tampering is detected through artifacts created by Color Filter Array or Cromatic Aberration during image creation process in most digital cameras.

There are other approaches that instead of using the lack of artifacts generated at image creation to detect tampering, rely on locating common techniques used during the tampering process, like smoothing [12], different resolution levels [3] or cloned zones over an image [8,10].

Some tampering detection methods, as Kee et al in [6], take into account the EXIF metadata contained in JPEG headers to extract useful information such as camera model.

In another way, Invariant Features methods like SIFT or SURF have also been used in tampering detection [9].

There are also some methods [4] that deal with our specific problem, blind splicing detection. It uses a different approach detecting double quantization examining the DCT coefficients. Our proposal is faster, has been tested with larger datasets and offers higher success rates.

We present a method based on JPEG ghosts by Farid [3] to detect whether an image has been tampered by splicing or not. See figure 1 for an example of spliced image. Furthermore, we trained a Support Vector Machine (SVM) algorithm in order to distinguish between this kind of tampered images and original images. Our proposed feature vector for SVM training is made up by the error variations between an image and its re-compressions at different quality levels.

The rest of the paper is organized as follows: In section 2 our proposed method is described. Experiments carried out and results obtained are introduced in section 3. Finally, the conclusions of our work are presented in section 4.

2 Features Extraction from Error Variation

Following Farid's proposal introduced in [3], if we convert a block of an image to frequency space by using a 2D DCT (Discrete Cosine Transform), we obtain a set of coefficients c quantized by an amount q. If we consider the coefficients c_0 quantized by an amount q_0 and then we quantize them again with an amount $q_1 < q_0$ we obtain the coefficients c_1. Quantizing the coefficients c_1 by q_2 provides c_2. The difference between c_1 and c_2 will be minimal when $q_2 = q_1$ but in this case, we will find a second minimum when $q_2 = q_0$ due to the previous quantization of the coefficients by q_0. Farid called this second minimum a JPEG ghost, because it reveals that the coefficients were previously quantized.

As Farid suggested, instead of computing this difference between the quantized DCT coefficients, we can compute it directly from the pixel values as in Equation (1).

$$d(x, y, q) = \frac{1}{3} \sum_{i=1}^{3} [f(x, y, i) - f_q(x, y, i)]^2 \tag{1}$$

Fig. 1. Left: Original image. Right: Tampered (spliced) image used as example. A chicken has been pasted on the ground.

where each d(x,y,q) stores the subtraction image (Error Image) between an original image f(x,y,i) on its three 'i' RGB channels and 'x' and 'y' coordinates and its re-compression $f_q(x, y, i)$ at quality 'q'.

In order to create a feature vector which is similar for images of the same class while dissimilar between the two classes, we followed the outline shown in Figure 2. We contemplated two classes: tampered and non-tampered images. First, the original image is re-compressed to different qualities from q = 60 to 90 with step 2. Then, the differences between the original image and its re-compressions are computed following Equation 1. The result of this process is a set of 16 Error Images (Figure 3).

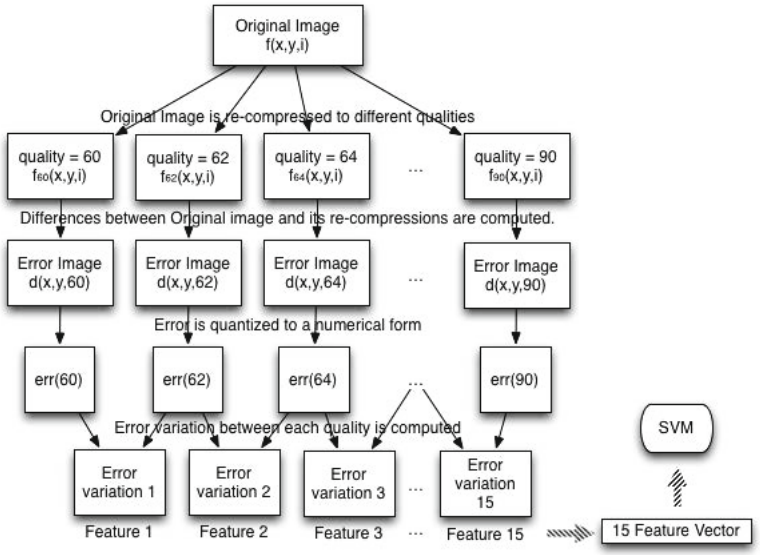

Fig. 2. This chart represents the features extraction process. These features make up the feature vector used for the SVM training.

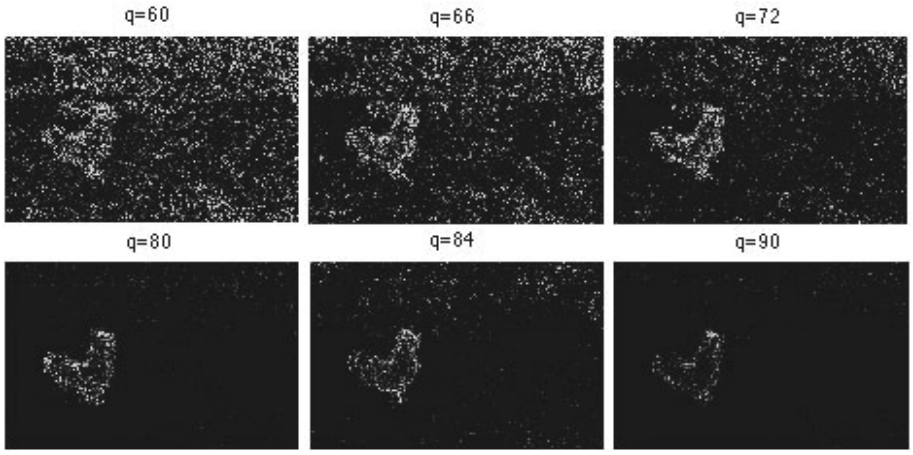

Fig. 3. White dots represent differences between the tampered image (Figure 1 Right) and its re-compressed image at given quality 'q'. As it can be observed, the original image has a quality level about 80 because the Error Image at q=80 shows the lower quantity of white dots on the background. There is also a central region of white pixels (where the chicken stands) that keeps being present for all Error Images, meaning that the central region is not compressed at any of those qualities (from 60 to 90) or presents a missaligntment with the rest of the image. In any case, we can affirm that our image has been tampered because there is not a single compressed image at a given quality similar enough to our image.

It is now when our approach differs from [3]. Farid considered a spatially averaged and normalized difference measure in order to reduce texture fluctuations due to the underlying image content. Also in his work, the two-sample Kolmogorov–Smirnov (K–S) statistic together with a threshold value is employed to classify the images as tampered or not. On the contrary, we skip the normalization step and prove that our method is both more robust and simpler. Furthermore, our approach finds a distinctive feature vector from the non-normalized differences which will be used to train an automatic and robust SVM algorithm.

Once the Error Images are obtained, the error is quantified to a numerical value by summing all the pixel values on each Error Image (Equation 2). This process results in a set of 16 values, one for each Error Image. Finally, error variations are computed by obtaining the differences between two consecutive quantified errors. These 15 error variations are actually the features which form the feature vector used in the SVM classification.

$$err(q) = \sum_{x=1}^{width} \sum_{y=1}^{height} (d(x, y, q)) \tag{2}$$

For non-tampered images (Figure 4 Left), the error will be decreasing as the quality of the re-compressed images approaches the quality of the original image. If exits a re-compressed image at the same quality as the original one, this error

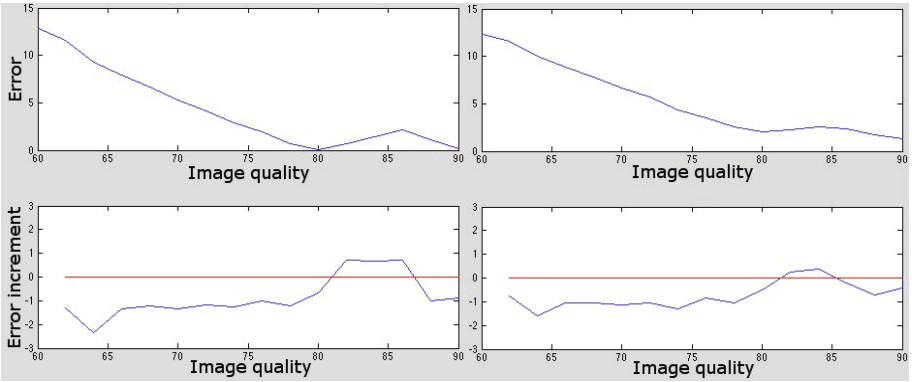

Fig. 4. Upper graphs represent the error level along the different qualities whereas lower graphs represent the variation of the error between each pair of re-compressed images at consecutive quality levels. Left: Non-tampered image error graph (Figure 1 Left). Given that the original image is compressed at a quality around 80, the error of the original image and the re-compressed image with quality equals to 80 falls near 0. Right: Spliced image error graph corresponding to Figure 3. As we can observe, the error reduces while approaching at quality = 80 (quality of the biggest region of the image) but not reaching zero, due to the presence of some error pixels introduced by the spliced region from a different quality or misalignment. We can observe that the error decreases again while approaching at quality 90, this is probably because at highest qualities an image is practically unaltered by compression.

will drop to zero. When the image has been tampered by splicing a region from an image with a different quality (Figure 4 Right), the error will be never close to zero at any given quality 'q' since there are regions on the image compressed at a quality different than 'q' or misaligned that will contribute to the error. Moreover, this error will decrease slower than in the case of non-tampered images due to the lack of an exact matching, hence the variation of the error is smoother. We could have just used the error values to extract the features for the SVM training, but instead we chose the error variations since it informs about how quick the compressed images approach to the most similar one to the original.

Given that most of the images are compressed at high qualities, most of error graphs, originals or tampered (Figure 4), will start with a high error value at lower qualities which will be decreasing at higher qualities.

Here we present another example. On the one hand, let us suppose that we have an image called 'A' compressed at quality 82. If we obtain its re-compressions at qualities from 60 to 90 with step 2, there will be an image (named 'AQ82') that will be exactly equal to Image 'A'. Hence the difference between 'A' and 'AQ82' will be zero and the error curve will reach value zero at that point. On the other hand, assume we have an image called 'C' composed by two regions. One from image 'A' at quality 82 and the other one from an image 'B' compressed at quality 68. If we obtain the re-compressions of 'C' from 60 to 90 with step 2, there will no be re-compressed images exactly equal to 'C' so the

error curve will never be equal to zero. However, errors of image 'C' with respect to image 'CQ82' and image 'CQ68' will be small (local minimum). Moreover, error between 'C' and 'CQ82' (respectively with 'C' and 'CQ68') will be smaller when larger regions of image B are considered.

3 Experiments and Results

To validate our proposal, two experiments were carried out.

The first one was aimed to compare results with Farid's method [3]. We used 183 untampered images from Columbia Uncompressed Image Splicing Detection Dataset [5]. We created the dataset following the process detailed in Farid's work in order to make our comparison as accurate as possible. First, our images were resized to 512×384 pixels and saved with a quality of 90 and for each one, a squared region of 50×50, 100×100, 150×150 and 200×200 pixels was extracted and saved with different qualities of 65, 70, 75, 80 and 85. Then, these patches were reinserted into its original image resulting into a set of 2940 tampered images with different tampered region sizes and qualities. SVM was trained using our proposed feature vectors and results were measured using k-fold cross-validation (Randomly distributing our dataset into $k = 10$ disjoint subsets, using 9 of them for training and the remaining one for testing. Repeating this process 10 times, selecting a different testing subset each time). Results, presented in table 1, show that our method is robust to region size and differences of quality. It is worth to mention that an error rate of 0.00% was achieved when classifying non-spliced images using this dataset. Farid's results are shown in the right side of the table for comparison purposes. His proposal offers better performance when the quality difference between the tampered region and the rest of the image is large, but is really poor when dealing with small regions or regions with similar quality levels. Our proposal greatly outperforms Farid's at those situations.

For our second experiment we used CASIA1 Tampered Image Detection Evaluation Dataset leading to a more general test. 700 non-tampered and 700 tampered images were used from this dataset. Tampered images were created

Table 1. Left: Hit rates yielded by our proposal. Right: Hit rates from Farid's work using a similar dataset. Several pasted region sizes and quality differences between the original images and the pasted regions ranging from 5 to 25 were considered.

Pasted size (pixels)	**Quality differences between the original and the pasted regions**									
	Proposed method					**Farid's method [3]**				
	5	10	15	20	25	5	10	15	20	25
200x200	97.6%	97.6%	97.6%	98.3%	97.6%	14.8%	52.6%	88.1%	93.8%	99.9%
150x150	97.2%	97.2%	97.3%	98.2%	97.3%	14.1%	48.5%	83.9%	91.9%	99.8%
100x100	97.9%	97.3%	97.6%	98.3%	97.9%	12.6%	44.1%	79.5%	91.1%	99.8%
50x50	97.3%	97.6%	97.3%	97.6%	97.6%	5.4%	27.9%	58.8%	78.8%	97.7%

by splicing, copying and pasting regions from one non-tampered image to another. Results were also obtained using k-fold cross-validation (k=10) and Least Squares SVM method. This test achieved an overall hit rate of 98.71% and just a 0.44% of non-tampered images were classified as tampered.

4 Conclusion

We have proposed a method for tampering detection on images containing spliced regions at different qualities. Our proposal obtains a high hit rate (98.71%) and it is robust against tampered region shape, alignment, size, and quality differences which proves its effectiveness in order to automatically detect tampered images. Two experiments were carried out. The first one aimed to compare results with Farid's [3] proposal. Our method offers several advantages like invariance against quality differences, shape and size of the tampered region. We widened the scope of the second experiment by using CASIA1 dataset obtaining just a 0.44% of false positives. This work also provides an automatic detection method using machine learning with an average computational time of 0.38 seconds for each image (time was measured on an Intel Core 2 Duo 2.4 GHz, 4GB 1067 MHz DDR3), including feature extraction and classification time. The high precision obtained detecting non-spliced images (99.56%) makes our proposal very useful in forensic science scenarios such as courts where it is highly important to know the veracity of an image.

Acknowledgment. This work has been supported by the ASASEC (www.asasec.eu) European Project with reference HOME/2010/ISEC/AG/043, by grants DPI2012-36166 and via the pre-doctoral FPU Spanish Government fellowship program.

Credits for the use of the CASIA Image Tempering Detection Evaluation Database (CAISA TIDE) V1.0 are given to the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Science, Corel Image Database and the photographers. http://forensics.idealtest.org.

References

1. Dirik, A.E., Bayram, S., Sencar, H.T., Memon, N.: New features to identify computer generated images. In: IEEE International Conference on Image Processing, ICIP 2007, September 16-October 19, vol. 4, pp. IV-433–IV-436 (2007)
2. Dirik, A.E., Memon, N.: Image tamper detection based on demosaicing artifacts. In: 2009 16th IEEE International Conference on Image Processing (ICIP), pp. 1497–1500 (November 2009)
3. Farid, H.: Exposing digital forgeries from jpeg ghosts. IEEE Transactions on Information Forensics and Security 4(1), 154–160 (2009)
4. He, J., Lin, Z., Wang, L., Tang, X.: Detecting doctored JPEG images via DCT coefficient analysis. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 423–435. Springer, Heidelberg (2006)

5. Hsu, Y.-F., Chang, S.-F.: Detecting image splicing using geometry invariants and camera characteristics consistency. In: 2006 IEEE International Conference on Multimedia and Expo, pp. 549–552 (July 2006)
6. Kee, E., Johnson, M.K., Farid, H.: Digital image authentication from JPEG headers. IEEE Transactions on Information Forensics and Security 6(3), 1066–1075 (2011)
7. Li, C., Hong, L.: Adaptive fragile watermark for image authentication with tampering localization. In: 2nd International Conference on Anti-counterfeiting, Security and Identification, ASID 2008, pp. 22–25 (August 2008)
8. Li, W., Yu, N.: Rotation robust detection of copy-move forgery. In: 2010 17th IEEE International Conference on Image Processing (ICIP), pp. 2113–2116 (September 2010)
9. Li, Y.: A robust forensic method based on scale-invariance feature transform. In: 2011 International Conference on Multimedia Technology (ICMT), pp. 5246–5249 (July 2011)
10. Luo, W., Huang, J., Qiu, G.: Robust detection of region-duplication forgery in digital image. In: 18th International Conference on Pattern Recognition, ICPR 2006, vol. 4, pp. 746–749 (2006)
11. Valenzise, G., Tagliasacchi, M., Tubaro, S., Cancelli, G., Barni, M.: A compressive-sensing based watermarking scheme for sparse image tampering identification. In: 2009 16th IEEE International Conference on Image Processing (ICIP), pp. 1265–1268 (November 2009)
12. Ying, C., Yuping, W.: Exposing digital forgeries by detecting traces of smoothing. In: The 9th International Conference on Young Computer Scientists, ICYCS 2008, pp. 1440–1445 (November 2008)
13. Hu, Y.-P., Han, D.-Z.: An svd-based self-embedding watermarking method for image authentication. In: 10th IEEE Singapore International Conference on Communication Systems, ICCS 2006, pp. 1–5 (October 2006)
14. Zhang, P., Kong, X.: Detecting image tampering using feature fusion. In: International Conference on Availability, Reliability and Security, ARES 2009, pp. 335–340 (March 2009)