Detecting Resized Double JPEG Compressed Images – Using Support Vector Machine

Hieu Nguyen, Stefan Katzenbeisser

Detecting Resized Double JPEG Compressed Images - Using Support Vector Machine

Hieu Cuong Nguyen and Stefan Katzenbeisser

Computer Science Department, Darmstadt University of Technology, Germany
{cuong,katzenbeisser}@seceng.informatik.tu-darmstadt.de

Abstract. Since JPEG is the most widely used compression standard, detection of forgeries in JPEG images is necessary. In order to create a forged JPEG image, the image is usually loaded into a photo editing software, manipulated and then re-saved as JPEG. This yields to double JPEG compression artifacts, which can possibly reveal the forgery. Many techniques for the detection of double JPEG compressed images have been proposed. However, when the image is resized before the second compression step, the blocking artifacts of the first JPEG compression are destroyed. Therefore, most reported techniques for detecting double JPEG compression do not work for this case. In this paper, we propose a technique for detecting resized double JPEG compressed (called RD-JPEG) images. We first identify features that can discriminate RD-JPEG images from JPEG images and then use Support Vector Machines (SVM) as a classification tool. Experiments with many RD-JPEG images with different quality and scaling factors indicate that our technique works well.

Keywords. SVM, classification, image forensics, re-sampling

1 Introduction

Due to the large number of available image processing tools, digital images can easily be altered without leaving visual evidence. Therefore, developing techniques for judging the authenticity of digital images became an urgent need. There are many types of image forgeries, which can be detected by different image forensic methods [1]. Since JPEG is the most popular image type and it is supported by many applications, it is worthwhile to develop forensic techniques for JPEG images.

Although there are many ways of making forgeries in a JPEG image, most share three main steps: 1) loading the JPEG image which is compressed by quality factor QF_1 to a photo editing software, 2) manipulating this image and 3) re-compressing it as a JPEG file with quality factor QF_2. Consequently, the forged image is doubly JPEG compressed (called D-JPEG). Detecting artifacts of double JPEG compression is an important step to judge whether a JPEG image is authentic. To this end, several techniques have been developed [2–8]. The authors in [2, 3] found that when QF_1 is different from QF_2, periodic artifacts are present in the histograms of the DCT coefficients of D-JPEG images. The periodicity can be recognized in the Fourier transform through peaks in the spectrum. Lin et al. [4] expanded the global approach of [3] by locating the tampered regions in D-JPEG images. Bianchi et al. [5] proposed an enhanced version of [4], leading to an improvement of the accuracy of the algorithm. The authors in [6, 7] showed that the distribution of the most significant digit of the DCT coefficients in JPEG images follows the generalized Benford distribution. This distribution is very sensitive to double JPEG compression and this property can be applied to detect D-JPEG images. Chen et al. [8] proposed a set of image features, which have subsequently been evaluated by a SVM based classifier.

A limitation of these techniques is that they cannot detect D-JPEG images if the JPEG images are cropped before the second compression step is applied. The reason is that the corresponding blocking grids in the first compression and in the second compression are no longer aligned. In order to overcome this limitation, some other techniques have been proposed [9–11]. In [9] a blocking artifact characteristic matrix (BACM) is computed to measure the symmetric representation of the blocking artifacts introduced by JPEG compression. Since the symmetry of the BACM of a JPEG image is destroyed after the image is cropped, the BACM can be used as evidence for detecting cropped double JPEG compressed images. The authors of [10] model the linear dependency of the "within-block" pixels (pixels that are not on the border of segmented 8×8 image blocks), compute the probability of the pixel being linearly correlated to its neighbors and form the map of the probabilities of all pixels in the image. The map is converted to Fourier domain and several statistical features from the different peak energy distribution are extracted in order to discriminate cropped D-JPEG images from non-cropped D-JPEG images. A simple yet reliable technique to detect the presence of cropped double JPEG compression has been introduced in [11]. This technique is based on the observation that the DCT coefficients exhibit an integer periodicity when they are computed according to the grids of the primary JPEG compression.

Although [9–11] work well for detecting cropped double JPEG compressed images, they are defeated if the images are resized before the second compression. The reason is that due to the effect of re-sampling, the blocking artifacts will be broken. The authors of [12] demonstrated the influence of resizing on the detection results of [7, 8]. To the best of our knowledge, there are only a few techniques for detecting resized double JPEG compressed (RD-JPEG) images [13–15]. Kirchner and Gloe [13] apply a re-sampling detection technique (which was originally designed to work with uncompressed images) to JPEG images and analyze how the JPEG compression affects the detection output. A limitation of [13] is that the detection rates when applied to RD-JPEG images are very low if QF_1 is much larger than QF_2. Besides, if the JPEG images are down-sampled before the second compression, the technique is mostly defeated. The technique [14] extracts neighboring joint density features and applies SVM to them. Although this technique works for both up-sampled images and down-sampled images by different interpolation methods, it is analyzed by the authors only for quality factors (both QF_1 and QF_2) of 75 and no information on false positives is given. Bianchi and Piva [15] proposed an algorithm, which can be summarized by some steps: 1) estimate the candidate resizing factor; 2) for each candidate factor, undo the image resizing operation and measure the NLDP (near lattice distribution property); 3) if the result is greater than a predefined threshold, label the image as resized double JPEG compressed. Furthermore, the technique [15] can estimate both the resize factor and the quality factor of the first JPEG compression of the analyzed image. The experimental results in [15] show that it surpasses [13] in the same test condition, but similar to [13], it seems more difficult to detect when QF_1 is much larger than QF_2.

In this paper, we propose a new technique to detect RD-JPEG images. The technique first reveals specific features of JPEG images by using a re-sampling detector. These features are subsequently fed to SVM-based classifiers in order to discriminate RD-JPEG images from JPEG images. In comparison to [13], our technique does not require to distinguish in detail the peaks caused by JPEG compression from the peaks caused by re-sampling. In comparison to [14], our approach does not need to extract complex image features for classification. The technique [15] consists of some intricate steps, which mostly use for the purpose of reverse engineering of resized double JPEG compressed images.

In Section 2, we briefly introduce state-of-the-art re-sampling detection techniques. Re-sampling detection is an important step of our technique and any of the mentioned re-sampling detectors can be used in our construction. The proposed detection algorithm for RD-JPEG images is explained in Section 3 and experimental results are shown in Section 4. Lastly, the paper is concluded in Section 5.

2 Techniques for Image Re-sampling Detection

To create a convincing forged image, the geometry of the image or some portions of it is often transformed. Once a geometric transformation (such as resizing or rotation) is applied to an image, a re-sampling process is involved. Interpolation is the central step of re-sampling in order to estimate the value of the image signal at intermediate positions to the original samples. Based on specific artifacts created by interpolation, there are several techniques to detect traces of re-sampling in digital images.

Gallagher [16] realized that low-order interpolated signals introduce periodicity in the variance function of their second derivatives. Based on this observation, the author proposed a technique to detect whether an image has been re-sampled. A limitation of this technique is that it works only in the case of image resizing. Using the Radon transform, Mahdian and Saic [17] improved [16] so that their technique can detect not only image resizing, but also image rotation. Popescu [18] noted that there are linear dependencies between neighboring pixels in re-sampled images. These correlations can be determined by using the Expectation/Maximization (EM) algorithm. The output of the algorithm is a matrix indicating the probability of every image sample being correlated to its neighbors (called p-map). The p-map of a re-sampled image usually contains periodic patterns, which are visible in the Fourier domain.

A drawback of [18] is that its computational complexity due to the use of the EM algorithm. Based on [18] some improved techniques have been introduced in [19] and [20]. The author in [19] showed that the p-map of a re-sampled image is periodic and this periodicity does not depend on the prediction weights that are used to compute the correlations of neighboring pixels. Therefore, he used a predefined set of prediction weights to compute the correlation probability of every image sample and designed a fast re-sampling detection technique which bypasses the EM algorithm.

Although the values of prediction weights do not affect the periodicity of the p-map in theory, the authors in [20] found that the selected set of predefined weights can strongly affect the obtained results: using one predefined set of weights for detection, peaks can be recognized in the transformed p-map, but using another set, peaks are not evident (though the periodicity exists in theory). Therefore, they use a predefined set, which is chosen through experimentation and apply the Radon transformation to the probability map of the analyzed image in order to enhance the frequency peaks and consequently the robustness of the overall technique. An example for detection of (uncompressed) re-sampled images by using [20] is presented in Fig. 1.

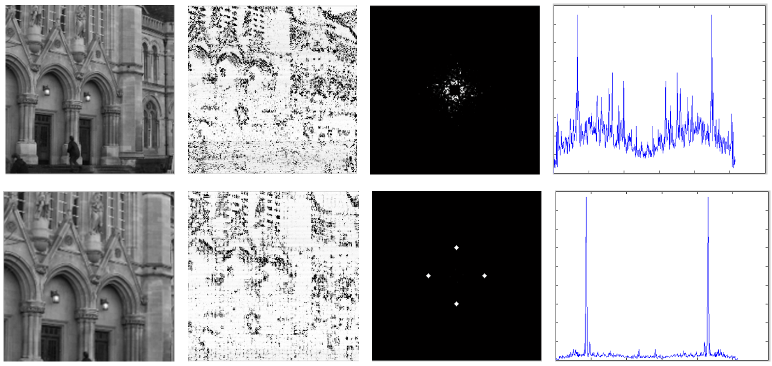


Fig. 1. Shown in the top row is the original image and shown in the bottom row is the re-sampled image by a factor of 1.2. Shown in the left most column are the original image and the re-sampled image. Shown in the middle columns are the p-maps and the magnitudes of the Fourier transforms of the p-maps. Shown in the right most column are the Fourier transformations of the Radon transforms of the p-maps.

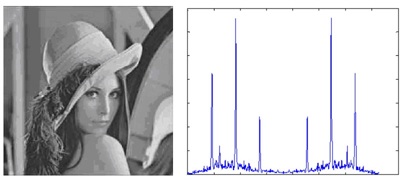


Fig. 2. Shown in the left is a JPEG image of Lena and in the right is the detection result by using a re-sampling detection technique.

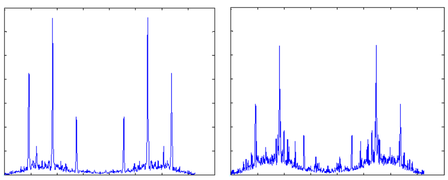


Fig. 3. Shown in the left is the detection result of the JPEG image of Lena and in the right the detection result of the RD-JPEG version of the same image.

The techniques [16–20] work well for detecting traces of re-sampling in uncompressed images. However, they fail when applied to JPEG images. The reason is that JPEG compression has an effect similar to nearest neighbors interpolation and the re-sampling detector will get confused [16]. An example can be seen in Fig. 2, which shows a JPEG image that has not been re-sampled, yet the spectrum when applying the re-sampling detector [17] contains strong peaks.

In the next section, we propose a technique which uses one of these re-sampling detectors as the first step for detecting RD-JPEG images. Although any mentioned re-sampling detection technique in this section can be used, we choose [17] because of its efficiency as well as its speed.

3 Proposed Technique for Detecting RD-JPEG Images

When using [17] to detect re-sampling in both JPEG images and RD-JPEG images, we empirically found that the detection results of RD-JPEG images seem to have more peaks than those of JPEG images. An example is shown in Fig. 3. This is because the detection result of a RD-JPEG image contains not only the peaks introduced by JPEG compression, but also the peaks due to re-sampling. Nevertheless, the difference is not always easy to recognize by human eyes. Besides, it is necessary to automatically classify RD-JPEG images from JPEG images. To this end, we first apply the technique [17] to JPEG images, and then extract the values of maximal peaks from the normalized Fourier spectrum. The extracted features are subsequently fed to SVM-based classifiers in order to discriminate RD-JPEG images from JPEG images. Since SVM is only a binary classifier, we use two different approaches to design SVM classifiers for detecting RD-JPEG images.

In the first approach, we design a single SVM classifier for directly distinguishing JPEG and RD-JPEG images, compressed by different quality factors. To this end, the features of a set of JPEG images and their re-sampled versions (the number of JPEG and re-sampled JPEG images are the same) are extracted for training a SVM classifier. This approach is simple and suitable for many situations in practice when we do not know the quality factors of the analyzed images. However, through experiments, reported in Section 4, we find that this technique works well mostly when QF_1 is lower than the QF_2.

The second approach is based on the idea that while QF_1 of a double JPEG compressed image is usually not known to the analyst, QF_2 can reliably be computed from the bitstream of the JPEG image (see Appendix A). Thus, instead of using one single classifier for all quality factors, we design several different SVM classifiers, each of which distinguishes JPEG and RD-JPEG images for one specific value of QF_2. Once the last quality factor of an analyzed JPEG image is known, the corresponding classifier will be applied to it. The method to design a classifier for a particular QF_2 is similar to the first approach: we first use a set of JPEG images and another set of RD-JPEG images (the numbers of images in both sets are the same and every image is compressed by QF_2) and then extract image features for training. In other words, the last quality factor of a tested image is first identified, and then the image will be analyzed by the corresponding qualifier. In next section, we discuss experimental results for both approaches.

4 Experimental Results

First, we randomly choose 200 uncompressed images from the UCID image database [21]. We create 5 datasets of JPEG images by compressing the uncompressed images with the quality factors of 40, 50, 60, 70, and 80. The JPEG images are subsequently resized by a scaling factor of 1.2 and recompressed by different factors of 40, 50, 60, 70, and 80. As a result, we obtained 5 datasets of 1000 RD-JPEG images corresponding to each dataset of JPEG images.

To test the first approach, we create a single SVM classifier by using two groups of JPEG images and RD-JPEG images (with the scaling factor of 1.2) for training. After the training process (presented in Section 3) we apply the classifier to test RD-JPEG images. In training, we consider two cases of different quality factors: 1) 100 JPEG images compressed by a quality factor of 50 and 100 RD-JPEG images re-compressed by a quality factor of 70 (QF_1=50, QF_2=70 and scaling factor =1.2) and 2) 100 JPEG images compressed by a quality factor of 70 and 100 RD-JPEG images re-compressed by a quality factor of 80 (QF_1=70, QF_2=80 and scaling factor =1.2). Analyzing the detection results (see Table 1 and Table 2), we found that the technique works well for detecting RD-JPEG images where QF_1 is smaller than QF_2. Otherwise, when QF_1 is larger than QF_2, the detection rate is reduced. In our experiments, the false positive rates (computed by testing the classifier on datasets of JPEG images which have been compressed by different quality factors of 40, 50, 60, 70, and 80) are lower than 10% in the first case and lower than 8% in the second case.

In a more realistic scenario, we test the techniques on the RD-JPEG images, which have been resized with a different scaling factor than the factors are used in the training process. The datasets are created in the same way as above, except the scaling factor 1.1 is used instead of 1.2 (i.e. QF_1=70, QF_2=80 and scaling factor =1.1). Although the detection results (in Table 3) are clearly worse compare with Table 1 and Table 2, we found that the degradation is not significant; therefore, the technique can potentially work in case the scaling factor is unknown.

In the second approach, we consider 5 different cases corresponding to a QF_2 of 40, 50, 60, 70, and 80. The case of QF_2=40, we organize the training images into two groups: a group of 100 JPEG images (the quality factor of 40) and the other group of 100 RD-JPEG images (QF_1=50, QF_2=40 and scaling factor=1.2). The extracted features are used to train a SVM classifier that can be used to detect RD-JPEG images which compressed by the QF_2 of 40. We repeat this process for the other cases when QF_2 is 50, 60, 70, and 80. The detection results in testing RD-JPEG datasets are presented in Table 4. We noticed that following the second approach, the technique works well even if QF_1 is larger than QF_2. The false positive rates are lower than 10% (9%, 8%, 5%, 6% and 3% when testing JPEG images compressed by the quality factors of 40, 50, 60, 70, and 80 respectively). Since JPEG compression with a lower factor produces stronger peaks in the Fourier spectrum, it obtains higher false positives.

Table 1. Detection results using a single SVM classifier (training JPEG images compressed by QF=50 and RD-JPEG images re-compressed by QF_1=50, QF_2=70) for RD-JPEG images by the scaling factor of 1.2 and by different quality factors (QF_1 in rows and QF_2 in columns).

	40	50	60	70	80
40	65.5%	91.0%	99.5%	99.5%	84.5%
50	52.5%	80.0%	97.0%	99.0%	87.0%
60	35.5%	77.5%	92.5%	98.5%	88.0%
70	19.5%	67.5%	87.0%	99.0%	84.0%
80	10.5%	45.0%	79.5%	91.5%	78.0%

Table 2. Detection results using a single SVM classifier (training JPEG images compressed by QF=70 and RD-JPEG images re-compressed by QF_1=70, QF_2=80) for RD-JPEG images by the scaling factor of 1.2 and by different quality factors (QF_1 in rows and QF_2 in columns).

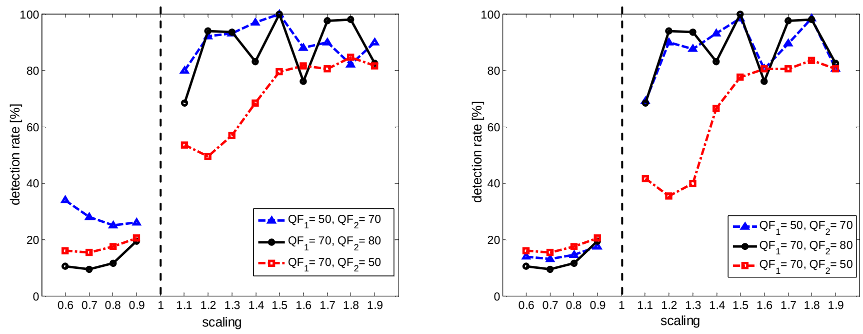
	40	50	60	70	80
40	70.0%	94.0%	98.5%	99.0%	95.0%
50	62.0%	80.0%	92.5%	98.5%	98.0%
60	48.0%	76.0%	87.5%	96.5%	99.0%
70	33.5%	68.0%	83.0%	93.5%	99.0%
80	24.0%	57.0%	69.0%	81.0%	92.0%

Table 3. Detection results using a single SVM classifier (training JPEG images compressed by QF=70 and RD-JPEG images re-compressed by QF_1=70, QF_2=80) for RD-JPEG images by the scaling factor of 1.1 and by different quality factors (QF_1 in rows and QF_2 in columns).

	40	50	60	70	80
40	37.0%	57.0%	63.5%	78.0%	82.5%
50	37.0%	58.0%	63.5%	78.5%	83.0%
60	26.0%	48.0%	66.5%	77.5%	87.0%
70	13.5%	43.5%	68.0%	77.5%	73.0%
80	10.5%	39.0%	62.5%	77.0%	86.5%

Table 4. Detection results using dedicated SVM classifiers for RD-JPEG images (depending on the quality factor of the second compression) by the scaling factor of 1.2 and by different quality factors (QF_1 in rows and QF_2 in columns).

	40	50	60	70	80
40	95.0%	91.5%	89.5%	99.0%	98.0%
50	90.0%	90.0%	88.5%	98.5%	99.5%
60	89.5%	91.0%	97.5%	98.0%	100%
70	87.5%	85.0%	95.0%	99.5%	98.0%
80	85.0%	80.0%	96.0%	100%	99.0%



(a) (b)

Fig. 4. Detection results for RD-JPEG images by different scaling factors: (a) the quality factors of the trained images and the test images are the same, (b) the quality factors of the trained images and the test images are different.

In order to assess the influence of scaling factor, we test the proposed technique for detection of RD-JPEG images with various scaling factors. The RD-JPEG images are created by resizing JPEG images (firstly compressed by QF_1) of different scaling factors (from 0.6 to 1.9) and then they are recompressed (by a different quality factor QF_2). We consider three cases: 1) QF_1=50 and QF_2=70, 2) QF_1=70 and QF_2=80 and 3) QF_1=70 and QF_2=50. We create different datasets of JPEG images and RD-JPEG images and in each case, the training and testing processes of the classifiers are conducted as described before. The detection results in various scaling factors are shown in Fig. 4a. Due to missing information in the down-sampling process, the detection rates of the down-sampled images are very low. Detecting up-sampled images is possible with much higher rates. In some cases, the detection rates even reach about 100%. In this scenario, the test images are compressed with the same quality factors as the training images (but with different scaling factor). We found that scaling factors affect the detection results: typically the detection rates tend to increase.

Lastly, in a more realistic scenario, we apply the technique trained by one image type (QF_1=70, QF_2=80, scaling factor = 1.2) to images with different types (QF_1=50 and QF_2=70, QF_1=70 and QF_2=50, and scaling factor ranges from 0.6 to 1.9). The detection results are presented in Fig. 4b. Although the results deteriorate (compare with Fig. 4a), we found that the degradation is not significant; therefore, the technique can potentially work in a real condition.

5 Conclusion

In this paper, we designed a technique for detecting resized double JPEG compressed images. The technique is based on applying a re-sampling detector to JPEG images, and extracting features from strong peaks of the normalized Fourier transformation. Then the extracted features are fed into a SVM-based classifier in order to discriminate RD-JPEG images from JPEG images. We propose two methods to design SVM classifiers: one single global classifier and several classifiers depending on the quality factor of the last compression. Although the first approach is simple and easy to use, the second approach achieves higher detection rates. In comparison with some existing techniques our technique has higher detection rates when the quality factor of the first compression is larger than the quality factor of the last compression and when detecting down-sampled images. We apply the technique to test RD-JPEG images resized with different scaling factors and found that the scaling factors can affect the detection results. In future, we will apply the technique for the detection of rotated double JPEG compressed images and use other re-sampling detectors in our technique so that we can compare their efficiency in the detector of RD-JPEG images.

References

1. Farid, H.: Image forgery detection. IEEE Signal Processing Magazine. 26, 16–25 (2009).
2. Lukáš, J., Fridrich, J.: Estimation of Primary Quantization Matrix in Double Compressed JPEG Images. In Proc. Digital Forensic Research Workshop. (2003).
3. Popescu, A.: Statistical Tools for Digital Image Forensics. PhD Thesis. (2004).
4. Lin, Z., He, J., Tang, X., Tang, C.-K.: Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis. Pattern Recognition. 42, 2492–2501 (2009).
5. Bianchi, T., Rosa, A. De, Piva, A.: Improved DCT coefficient analysis for forgery localization in JPEG images. ICASSP 2011. 2444–2447 (2011).
6. Fu, D., Shi, Y.Q., Su, W.: A generalized Benford's law for JPEG coefficients and its applications in image forensics. Proceedings of SPIE. 65051L–65051L–11 (2007).
7. Li, B., Shi, Y.Q., Huang, J.: Detecting doubly compressed JPEG images by using Mode Based First Digit Features. 2008 IEEE 10th Workshop on Multimedia Signal Processing. 730–735 (2008).
8. Chen, C., Shi, Y.Q., Su, W.: A machine learning based scheme for double JPEG compression detection. 2008 19th International Conference on Pattern Recognition. 1–4 (2008).
9. Luo, W., Qu, Z., Huang, J., Qiu, G.: A Novel Method for Detecting Cropped and Recompressed Image Block. IEEE ICASSP 2007. 217–220 (2007).
10. Chen, Y., Hsu, C.: Image Tampering Detection by Blocking Periodicity Analysis in JPEG Compressed Images. MMSP 2008. 803–808 (2008).
11. Bianchi, T., Piva, A.: Detection of Nonaligned Double JPEG Compression Based on Integer Periodicity Maps. IEEE Trans. on Information Forensics and Security. 7, 842–848 (2012).
12. Sutthiwan, P., Shi, Y.Q.: Anti-Forensics of Double JPEG Compression Detection. IWDW. 411–424 (2011).
13. Kirchner, M., Gloe, T.: On resampling detection in re-compressed images. WIFS. 21–25 (2009).
14. Liu, Q., Sung, A.H.: A new approach for JPEG resize and image splicing detection. Proceedings of the First ACM workshop on Multimedia in forensics - MiFor '09. 43 (2009).
15. Bianchi, T., Piva, A.: Reverse engineering of double JPEG compression in the presence of image resizing. 2012 IEEE International Workshop on Information Forensics and Security (WIFS). 127–132 (2012).
16. Gallagher, A.C.: Detection of Linear and Cubic Interpolation in JPEG Compressed Images. The 2nd Canadian Conference on Computer and Robot Vision (CRV'05). 65–72 (2005).

17. Mahdian, B., Saic, S.: Blind Authentication Using Periodic Properties of Interpolation. IEEE Transactions on Information Forensics and Security. 3, 529–538 (2008).
18. Popescu, A.C., Farid, H.: Exposing digital forgeries by detecting traces of resampling. IEEE Transactions on Signal Processing. 53, 758–767 (2005).
19. Kirchner, M.: Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue. Proceedings of the 10th ACM Workshop on Multimedia and Security - MM&Sec'08. (2008).
20. Nguyen, H.C., Katzenbeisser, S.: Robust Resampling Detection in Digital Images. CMS 2012. 3–15 (2012).
21. Schaefer, G., Stich, M.: UCID: an uncompressed color image database. Proc. SPIE, Storage and Retrieval Methods and Applications for Multimedia, San Jose, USA. 472–480 (2004).
22. Kornblum, J.: Using JPEG quantization tables to identify imagery processed by software. Digital Investigation. 5, S21–S25 (2008).
23. Chandra, S., Ellis, C.S.: JPEG Compression Metric as a Quality Aware Image Transcoding. Proceedings of USITS' 99. (1999).
24. Sallee, P.: Matlab JPEG Toolbox, http://www.philsallee.com/jpegtbx/index.html.

Appendix A. Determining the last quality factor of a JPEG images

The compression ratios of JPEG images are controlled by the quantization tables which used in the compression process. In this paper, we focus on images stored in the JPEG Interchange File Format (JFIF). The JFIF is the most commonly used format for JPEG data [22]. The quantization table that was used to compress an image is stored in the JFIF header [23]. This table (called *Ts*) can be identified by using the JPEG Toolbox [24].

The most commonly used standard quantization tables are published by the International JPEG Group (IJG). Based on the standard table (*Tb*), and the quality factor (*Q*), the quantization table can be computed as follows:

$$S = \begin{cases} \dfrac{500}{Q} & \text{if } Q < 50 \\ 200 - 2Q & \text{otherwise} \end{cases} \quad , \quad Ts[i] = \left\lfloor \frac{S * Tb[i] + 50}{100} \right\rfloor .$$

Conversely, when the tables Tb and Ts are known, the approximate value of the quality factor can be computed as follows [23]:

$$S' = \frac{Ts[i]*100 - 50}{Tb[i]} \quad , \quad Q' = \begin{cases} \left\lfloor \dfrac{200 - S'}{2} \right\rfloor & \text{if } S' \le 100 \\ \left\lfloor \dfrac{5000}{S'} \right\rfloor & \text{otherwise} \end{cases} .$$

Note that the function to predict the quality factor involves integer computation on the quantization table (*Ts*) that introduce integer rounding errors, so the value of *Q'* is close to *Q*. Following a suggestion in [23], then the computed quality factor (*Q'*) should be off by one or two.