Applied Computer Science, vol. 12, no. 3, pp. 17–28

Submitted: 2016-07-27
Revised: 2016-09-05
Accepted: 2016-09-09

Compression detection, Image quality, JPEG

Grzegorz KOZIEL[*]

PRIOR IMAGE JPEG-COMPRESSION DETECTION

Abstract

The paper presents two methods of prior JPEG-compression detection. In the first method the histogram of chrominance is analysed. In JPEG-compressed images the histogram contains significantly more local maxima than in uncompressed files. The second method is based on neighbouring pixel Ø value difference. In JPEG-compressed image the distribution of these values is different than the distribution counted on the edges of compression 8x8 blocks. These differences are summed up to create a classifier that allows to assess if the image was compressed.

1. INTRODUCTION

Steganography is the science of hiding information in another carrier called a container. A carrier can be an image. Further in the article we will focus on images, thus the container will be understood to be an image. In this case the data of the image chosen to hide data inside itself are modified in such a way as to hide a portion of information by including it directly in the image data. During this operation it is important to keep the image quality and not introduce detectable changes.

By detectable changes we understand any visual distortions that can be notified by observer or any statistical interference or characteristic artefacts in the image data. The problem is that in some containers it is much more difficult to meet these conditions. Especially when the image was previously compressed. After compression some characteristic features are introduced into the compressed image. If the steganographic method interferes with these

[*] Lublin University of Technology, Nadbystrzycka 38D, 20-618 Lublin, Poland,
phone: +48 81 538 46 08, e-mail: g.koziel@pollub.pl

features and changes the supposed characteristics of the previously compressed image, the steganalyst who examines the image to check for hidden data will be able to easily detect the presence of hidden information. Also the forgery of the previously compressed file can be detected by image features analysis.

1.1. Problem formulation

Because of problems which may occur while using a compressed image it is worth examining it for former compression. This is important especially when we do not know the file origin. Files downloaded from the Internet are very popular among users but they are often compressed. Even if the file is delivered to user in uncompressed form it could be converted from compressed format. This article discusses the problem of detection of prior JPEG compression. The following thesis was formulated: "It is possible, without a complex analysis, to detect if the analysed image was previously JPEG-compressed."

1.2. Background

The subject of prior image compression is widely discussed in the literature. This is important in steganography (Fridrich, 2005; Kodovsky & Fridrich, 2013; Lukáš & Fridrich, 2003; Pevny & Fridrich, 2008), image processing (Yang, Zhu & Huang, 2015) and forensics (Bianchi, Piva & Perez-Gonzalez, 2013; Fridrich, 2005; Piva, 2013; Popescu & Farid, 2005). In (Triantafyllidis, Tzovaras & Strintzis, 2002) a frequency-domain technique for image blocking artefact detection is proposed. The algorithm detects the image regions which contain visible blocking artefacts. The detection is done in the frequency domain. Blocking artefact inconsistencies are also used in (Ye, Sun & Chang, 2007) to detect digital forgeries of images. Authors of (Lin, Chang & Chen, 2011) propose the use of quantisation table estimation to measure the inconsistency among images to detect forgeries. The authors of (Liu & Bovik, 2002) propose a method of numerical assessment of the degree of artefact blocking in a visual signal. The method works in DCT domain. The steganalytic algorithm based on examining the compatibility of 8x8 pixel blocks with JPEG compression with a given quantisation matrix is presented in (Fridrich, Goljan & Du, 2001). In (Kodovsk`Y & Fridrich, 2013) the authors develop a JPEG-based steganalysis by examining the difference between an image with hidden data and an estimate of the cover image obtained by recompression with a JPEG quantisation table estimated from the stego image.

2. JPEG COMPRESSION

JPEG compression relies on the YCbCr colour model and discrete cosine transform (DCT). It contains the following stages (Przybyłowicz, 2008):

1. transformation to YCbCr colour model and resolution reduction,
2. image decomposition into blocks,
3. discrete cosine transform on each image block,
4. quantisation,
5. DCT matrix zigzag shift and Huffmann coding.

In the first stage RGB image representation is transformed into a YCbCr colour model. This means that each pixel will be described by three components: luminance (Y) and two chrominance components: anti-red (Cr) and anti-blue (Cb). Transformation from RGB model is done for each pixel separately according to Formula 1:

$$\begin{cases} Y = 0.299 * R + 0.587 * G + 0.114 * B \\ Cr = 128 - 0.168736 * R - 0.331264 * G + 0.53 * B \\ Cb = 128 + 0.5 * R - 0.4186688 * G - 0.081312 * B \end{cases} \tag{1}$$

where: R – value of red colour component,
G – value of green colour component,
B – value of blue colour component (Przybyłowicz, 2008).

After the transformation a vertical and horizontal resolution of chrominance components can be reduced because of lower human visual system (HVS) sensitivity to chrominance components than to luminance (Przybyłowicz, 2008).

An image prepared in this way is divided into blocks sized 8x8 pixels. Next, each block is independently processed with DCT and quantised. Quantisation is an operation during which each DCT coefficient is divided by a corresponding value taken from the predefined array Q. Each result is rounded to the closest integer. This operation allows for higher frequencies reduction. The human eye has a low sensitivity on quick luminance changes so higher frequencies can be reduced. At this stage the user can adjust the quality level (QL). The QL ranges from 1 to 100. Value 1 means the highest compression and the lowest quality. Value 100 means the best image quality and lowest compression level.

As a result of the reduction of the level of higher frequencies we obtain an array of coefficients, where some coefficients representing high frequencies are equal to zero. This array is changed into a vector by reading coefficients from the array in such a way as to place the low frequency coefficients at the beginning of the vector and the high frequency ones at the end of it.

Because of this at the end of the vector multiple zeros are placed. These zeros are replaced by the symbol EOB, which means that the rest of the vector is filled with zeros. This allows for image data size reduction. At the end the data are coded with Hufmann coding (Przybyłowicz, 2008). It is worth noticing that in the JPEG2000 compression format the cosine transform was replaced by a wavelet transform. Thanks to this a better compression can be obtained and an image transmitted through the Internet can be presented to the user in lower resolution before the whole image data are transferred. Today most of JPEG images are compressed in JPEG2000 format.

3. JPEG COMPRESSION DETECTION

Some images that can be download from the Internet look as if they are uncompressed. Unfortunately, there is a high chance that they were compressed earlier to store or transmit them effectively. Such images can be uncompressed before putting in the repository available for users. In the case of lossy compression the images keep the compressed format characteristics and some of their features. To prove this statement, a simple experiment was done. A raw picture (taken by Nikon D90, without any processing) was compressed and the differences between corresponding pixels in the original and the compressed file were calculated. In this way it was possible to determine how big the examined pixel's value change was. In this experiment a set of 10 different pictures were examined. Average values were calculated and presented in Figure 1. There we can see the results obtained after compression to JPEG format at various quality levels (25, 50 and 90). In the case of transforming raw image to JPEG a large number of pixels are changed. Only 20%–30% of pixels have the same value. The rest of them changes during compression. In Figure 1 only the value changes up to 22 were presented, but the biggest changes of pixel value are up to 50. Because of a small number of pixels having such a significant value change it was not possible to present them on the chart.

In the next step of the experiment the compressed image was uncompressed and saved in both bmp and png format. In this case the pixel value change was also examined according to the above-presented algorithm. In this experiment we found that all pixels had the same value after image format change. There were no pixels changing value. Changing format from JPEG to uncompressed form does not change pixel value – all the difference between the corresponding pixels is zero. This proves that no changes were introduced into the image during the examined process and all the characteristics and features of the compressed image are kept in its uncompressed form.

Each image has a set of various features that can be examined to check if the image was earlier compressed or not. It is not necessary to analyse all of them, only a selection of parameters, to detect if the image has any features that indicate prior compression. During the work over the problem in question various aspects were analysed, the final selection being:

– luminance and chrominance distribution,
– value differences between neighbouring pixels.

The presented research was prepared with a set of 280 images. Pictures were taken by Nikon D90 and written simultaneously in two formats: raw (NEF) and JPEG. The quality of JPEG files was various. As a result, two sets of images were collected. One set contained compressed files, the other uncompressed ones.

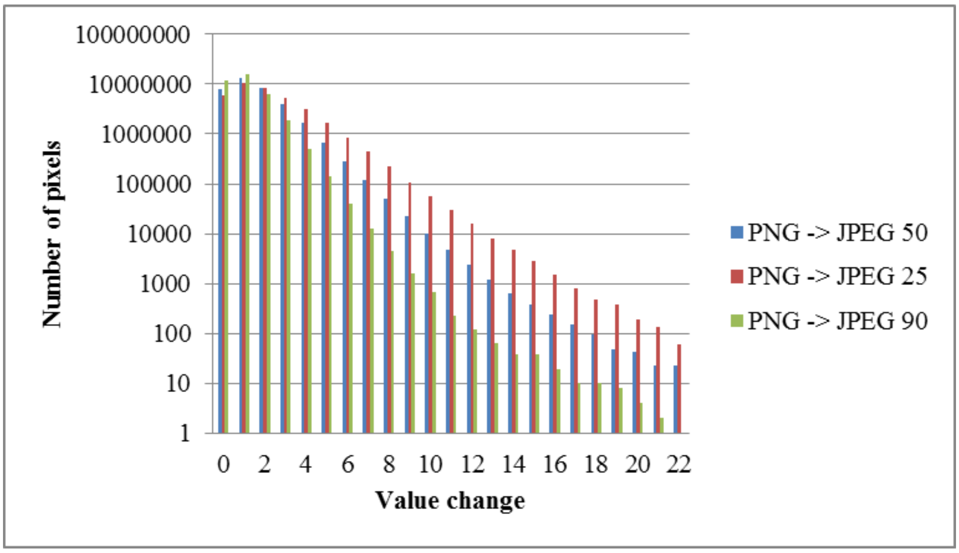

Fig. 1. Pixel value differences after image format change (own study)

3.1. Luminance and chrominance analysis

As mentioned in Section 2, during JPEG compression the colours are transformed to YCbCr model. The compression is done with the use of this colour model (Bianchi et al., 2013). This causes changes in the pixels value. To verify this the common histogram of luminance and chrominance components was prepared. The histogram obtained from the raw file is presented in Figure 2. Figure 3 presents the histogram obtained from the JPEG file.

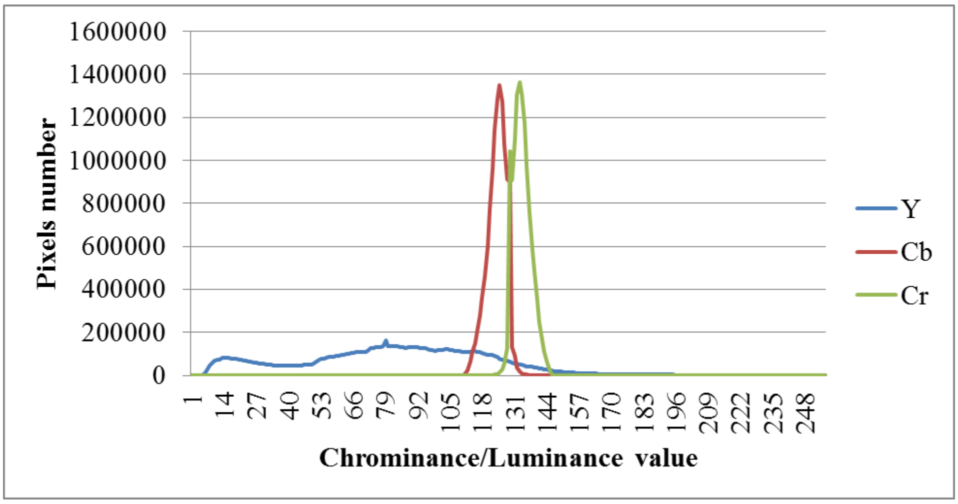

Fig. 2. Luminance and chrominance histogram for a raw image (own study)

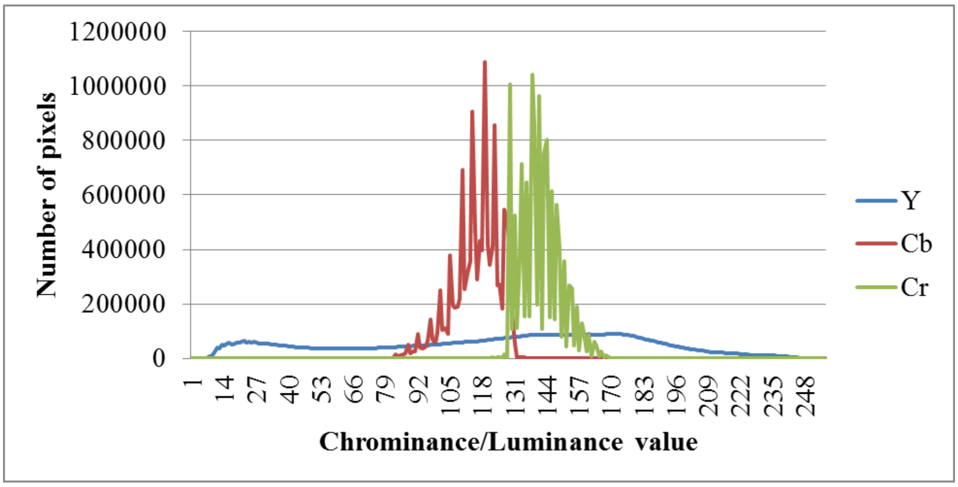

Fig. 3. Luminance and chrominance histogram for a JPEG image (own study)

Analysis of Figures 2 and 3 allows to notice a significant change of the chrominance components histogram. On the base of this change an attempt to create an image classifier was taken. The analysis of local maxima number in the luminance and chrominance components was done. The obtained results are presented in Figure 4.

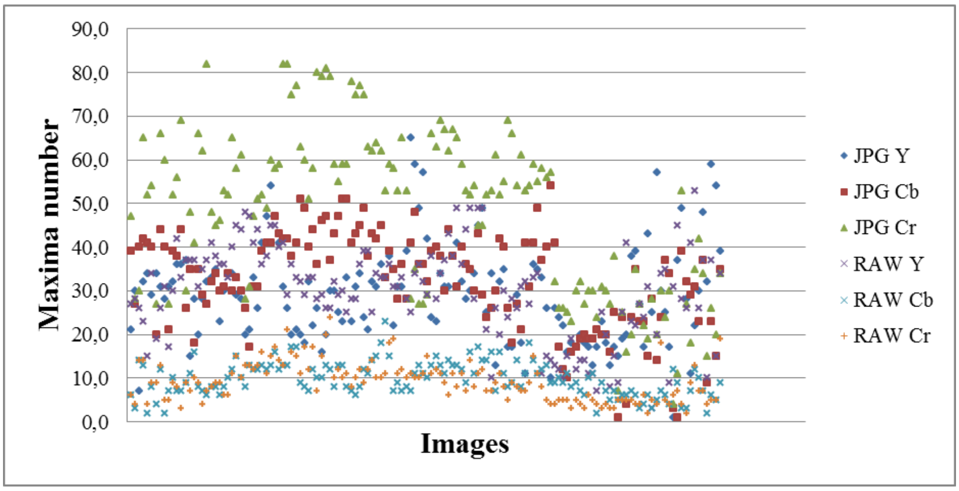

Fig. 4. Number of maxima in the luminance and chrominance histograms for JPEG and raw image (own study)

As we can see in Figure 4, there exists a significant difference between the number of chrominance maxima for JPEG and raw images. The number of maxima for JPEG images is higher. The numerical analysis of the obtained results was done as presented in Table 1.

Tab. 1. Numerical analysis of chrominance and luminance local maxima number (own study)

	JPEG			RAW		
	Y	Cb	Cr	Y	Cb	Cr
Biggest maxima number	65	54	82	53	23	24
Smallest maxima number	1	1	4	3	2	2
Biggest maxima number (VR=10%)	45	45	69	44	15	15
Smallest maxima number (VR=10%)	17	17	24	13	5	4
Biggest maxima number (VR=6%)	49	48	78	45	16	17
Smallest maxima number (VR=6%)	13	14	20	10	3	3

As can be seen, the number of maxima in chrominance histograms (Cb and Cr) should be bigger in the case of JPEG images. The biggest maxima number found in any image is presented in the first line of Table 1. Unfortunately, in some cases JPEG images have a small number of analysed maxima (the smallest numbers of maxima found in a JPEG image are presented in the second line of Table 2). The algorithm of finding a threshold for the Cr histogram is as follows:

1. count local maxima in the Cr histogram for JPEG images and put obtained values into table JPEG_Cr_max; sort the JPEG_Cr_max table in ascending order;
2. count local maxima in the Cr histogram for raw images and put obtained values into table RAW_Cr_max; sort the RAW_Cr_max table in descending order;
3. set the percent of values to remove VR=1%;
4. remove one percent of the first values from both tables;
5. check if the first value form the table RAW_Cr_max is smaller than the first value from the table JPEG_Cr_max; if yes, then return VR and first and last values from both tables; if not, then increase VR=VR+1% and go back to point 4.

According to the same algorithm the Cb histogram maxima were processed independently. The obtained results are presented in the last four rows of Table 1. These results show that for the component Cb the threshold should be set to 16 maxima (compare blue-marked cells). Each image having more maxima will be classified as JPEG-compressed. The accuracy of this classifier, according to the present research, is 90%, because 10% of values are out of the determined set. More precise results can be obtained with Cr component analysis. Cells form Table 1 marked orange show that there is still a big gap between raw and JPEG maxima number. Two last lines in Table 1 show results obtained for the set where only 6% of extreme results were removed. Cells marked green show that the JPEG files from the analysed set have in each case more maxima than the raw files. In this case the Cr component threshold can be set to 18 maxima. Each image having more maxima will be recognised as JPEG. The examined accuracy of this classifier is 94%.

3.2. Neighbouring pixel values difference

JPEG compression processes image data in blocks sized 8x8 pixels. Each block is processed independently. It causes an additional interference on the blocks edges. In the case of strong compression blocks could be visible as small squares having a slightly different colour than the neighbouring blocks. It is caused by a bigger value difference between border pixels coming from different blocks. This difference was examined and compared with the value differences among the pixels coming from the whole picture. The value difference between neighbouring pixels was counted in the RGB colour model according to the following algorithm:

1. create table RES for results having size 3x256; each row will keep results for another colour (R, G and B); initialise table RES with zeros;
2. set the read pixel index (PixIndex) to first pixel;

3. read first pixel from the image (marked ORG) and pixel placed below it (marked NGB);
4. for each colour:
   - read first pixel from the image (marked ORG) and pixel placed below it (marked NGB),
   - count index= |ValORG-ValNGB|,
   - increment RES[colour][index] by one;
5. set the read PixIndex to next pixel;
6. if the PixIndex indicates pixel from the image and not form the last line then go to point 3 of this algorithm;
7. count the pixel number (PixNum) in the image but the last line;
8. divide all values in RES table by PixNum;
9. return to RES table.

For the pixels placed on block edges value differences were counted according to the same algorithm, but ORG pixels were taken from the 8th line and next every eighth line. The RES table contains information about the percent of neighbouring pixels that differ about a defined value, for example the first cell in the first row keeps what percent of neighbouring pixels do not differ, the second cell tells what percent of neighbouring pixels differs about one, and so one.

As can be noticed, the differences between pixels are examined only vertically. It is possible to check them in other directions but it is not necessary because of good results obtained with the present method. Figure 5 presents the results obtained for the raw file, Figure 6 the results obtained for the JPEG. Symbols R, G, B mark differences among neighbouring pixels in the whole image. Symbols R edge, G edge, B edge mark differences among neighbouring pixels placed on the block edges. Series R difference, G difference, B difference are used to show the value differences between values obtained for the whole image and for the edges. These values were counted according to the formula 2:

$$\begin{cases} R_{difference} = R - R_{edge} \\ G_{difference} = G - G_{edge} \\ B_{difference} = B - B_{edge} \end{cases} \tag{2}$$

It can be noticed that the value differences between pixels are similar in the raw file, in contrast to the JPEG file. There, because of compression, differences are noticeable. On the basis of these differences a classifier was defined according to Formula 3:

$$C = \sum_{c=0}^{2} \sum_{i=0}^{N-1} \left| RES[c,i] \right|, N = 0..255 \tag{3}$$

where: c – colour index in the RES table,
N – value difference between pixels.

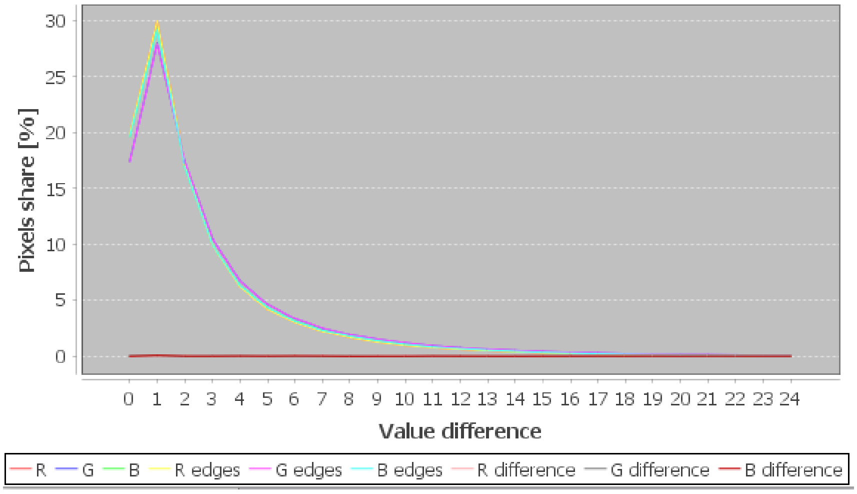

Fig. 5. Value differences between neighbouring pixels in a raw file (own study)

As can be seen in Figure 7, the classifier C has bigger values for JPEG images. The minimal classifier value found for the JPEG image is 5,01 whereas the biggest value found for the raw file is 1,77. The big gap between these results allows to define a treshold at the level of 3,38. Each image having a bigger value of the presented classifier will be recognised as JPEG-compressed. During tests all files were classified correctly by this classifier, so its acuracy is 100%.

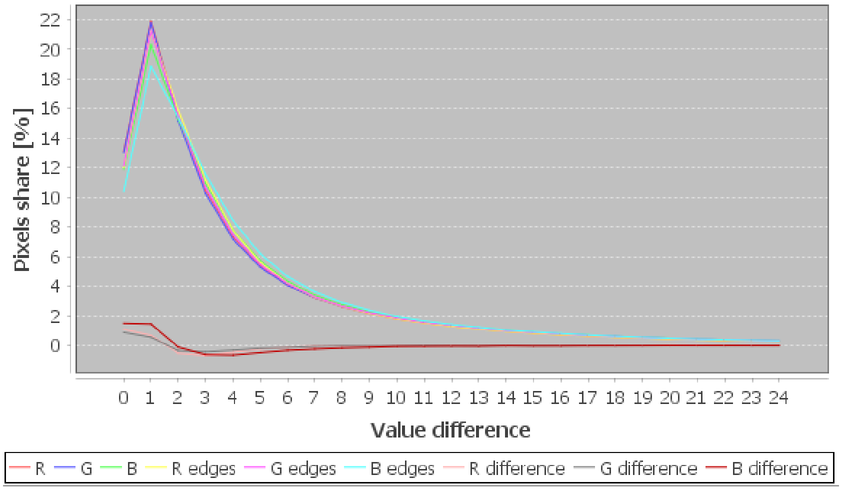

Fig. 6. Value differences between neighbouring pixels in a JPEG file (own study)

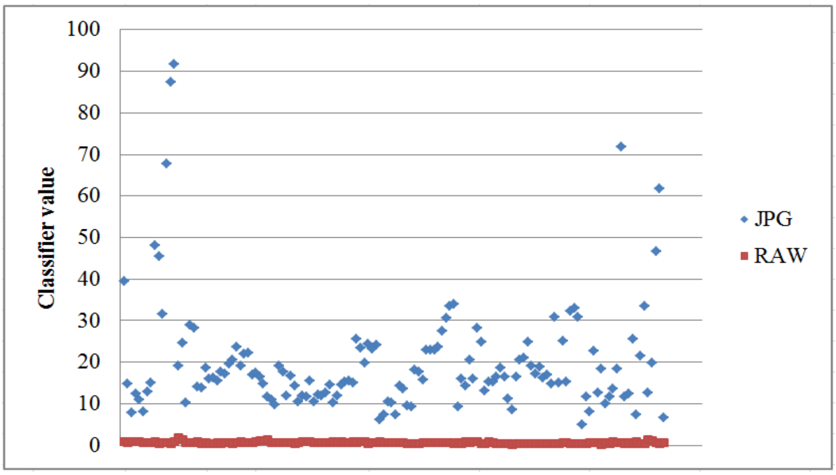

Fig. 7. Classifier values for JPEG and raw files (own study)

5. CONCLUSIONS

It is possible to efficiently detect prior JPEG compression with high accuracy. The methods presented recognise characteristic features introduced to the image during JPEG compression. The present article does not exhaust the topic. Future research will focus on frequency analysis and an attempt to build further classifiers will be made. The topic of prior compression detection is important in steganography, because previously compressed image usage will weaken the security of the applied protection. The prior potential container analysis allows for obtaining better results in steganography and ensures a higher level of hidden information security.

REFERENCES

Bianchi, T., Piva, A., & Perez-Gonzalez, F. (2013). Near optimal detection of quantized signals and application to JPEG forensics. *IEEE International Workshop on Information Forensics and Security (WIFS)*, 168-173. doi: 10.1109/WIFS.2013.6707813

Fridrich, J. (2005). Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes. *Information Hiding. 6th International Workshop*, *3200*, 67-81. doi: 10.1007/978-3-540-30114-1_6

Fridrich, J., Goljan, M., & Du, R. (2001). Steganalysis based on JPEG compatibility. In A. G. Tescher, B. Vasudev, & V. M. Bove (Eds.), *Multimedia Systems and Applications Iv* (vol. 4518, pp. 275-280). doi:10.1117/12.448213

Kodovsky, J., & Fridrich, J. (2013). JPEG-Compatibility Steganalysis Using Block-Histogram of Recompression Artifacts. *Information Hiding. 14th International Conference, 7692*, 78-93. doi: 10.1007/978-3-642-36373-3_6

Lin, G. S., Chang, M. K., & Chen, Y. L. (2011). A Passive-Blind Forgery Detection Scheme Based on Content-Adaptive Quantization Table Estimation. *IEEE Transactions on Circuits and Systems for Video Technology, 21*(4), 421-434. doi:10.1109/tcsvt.2011.2125370

Liu, S. Z., & Bovik, A. C. (2002). Efficient DCT-Domain blind measurement and reduction of blocking artifacts. *IEEE Transactions on Circuits and Systems for Video Technology, 12*(12), 1139-1149. doi:10.1109/tcsvt.2002.806819

Lukáš, J., & Fridrich, J. (2003). Estimation of primary quantization matrix in double com-pressed JPEG images. *Proc. Digital Forensic Research Workshop, 2*, 5-8.

Pevny, T., & Fridrich, J. (2008). Detection of double-compression in JPEG images for applications in steganography. *IEEE Transactions on Information Forensics and Security, 3*(2), 247-258.

Piva, A. (2013). An overview on image forensics. *ISRN Signal Process, 2013*, Article ID 496701, 22 pages. doi:10.1155/2013/496701

Popescu, A.C., & Farid, H. (2005). Statistical tools for digital forensics. *Information Hiding. 6th International Workshop, 3200*, 128-147. doi:10.1007/978-3-540-30114-1_10

Przybyłowicz, P. (2008). Matematyczne podstawy kompresji JPEG. *Centrum Modelowania Matematycznego Sigma*

Triantafyllidis, G. A., Tzovaras, D., & Strintzis, M. G. (2002). Blocking artifact detection and reduction in compressed data. *IEEE Transactions on Circuits and Systems for Video Technology, 12*(10), 877-890. doi:10.1109/tcsvt.2002.804880

Yang, J. Q., Zhu, G. P., Huang, J. W., & Zhao, X. (2015). Estimating JPEG compression history of bitmaps based on factor histogram. *Digital Signal Processing, 41*, 90-97. doi:10.1016/j.dsp.2015.03.014

Ye, S., Sun, Q., & Chang, E. C. (2007). Detecting digital image forgeries by measuring inconsistencies of blocking artifact. *IEEE International Conference on Multimedia and Expo*, 12-15. doi: 10.1109/ICME.2007.4284574