International Journal of Digital Crime and Forensics
Volume 11 • Issue 1 • January-March 2019

A Universal Image Forensics of Smoothing Filtering

Anjie Peng, School of Computer Science and Technology, Southwest University of Science and Technology, Mianyang, China & Guangdong Key Laboratories of Information Security Technology, Sun Yat-Sen University, Guangzhou China

Gao Yu, School of Computer Science and Technology, Southwest University of Science and Technology, Mianyang, China

Yadong Wu, School of Computer Science and Technology, Southwest University of Science and Technology, Mianyang, China

Qiong Zhang, School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, China

Xiangui Kang, School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, China

ABSTRACT

Digital image smoothing filtering operations, including the average filtering, Gaussian filtering and median filtering are always used to beautify the forged images. The detection of these smoothing operations is important in the image forensics field. In this article, the authors propose a universal detection algorithm which can simultaneously detect the average filtering, Gaussian low-pass filtering and median filtering. Firstly, the high-frequency residuals are used as being the feature extraction domain, and then the feature extraction is established on the local binary pattern (LBP) and the autoregressive model (AR). For the LBP model, the authors exploit that both of the relationships between the central pixel and its neighboring pixels and the relationships among the neighboring pixels are differentiated for the original images and smoothing filtered images. A method is further developed to reduce the high dimensionality of LBP-based features. Experimental results show that the proposed detector is effective in the smoothing forensics, and achieves better performance than the previous works, especially on the JPEG images.

KEYWORDS

Autoregressive Model, Digital Image Forensics, Local Binary Pattern, Smoothing Filtering

1. INTRODUCTION

The increase of forged images on the Internet has attracted much concern from researchers on multimedia security. When a forger creates a forged image, he often conducts smoothing filtering operations to beautify the forged image and make it look like an ordinary one. Thus, the forensics of smoothing filtering is able to provide auxiliary clues to identify the forged images. Furthermore, the smoothing filtering history of an image is an essential element for stegography (Kodovský & Fridrich, 2014; Pevný, Bas, & Fridrich, 2010) and steganalysis (Barni, Cancelli, &Esposito, 2010). Kodovský et al. pointed out thatit is not secure to embed a message into a smoothing filtered image (Kodovský

DOI: 10.4018/IJDCF.2019010102

& Fridrich, 2014). Therefore, the forensics of smoothing filtering is of particular significance in multimedia security.

There are some excellent works about the median filtering forensics (Yuan, 2011; Zhang, Li, Wang, & Shi, 2014; Chen, Ni, & Huang, 2013; Cao, Zhao, Ni, Yu, & Tian, 2010; Niu, Zhao, & Ni, 2017; Kang, Stamm, Peng, & Liu, 2012; Yang, Ren, Zhu, Huang, & Shi, 2017). Relatively, only a few universal forensic methods are devoted to detecting commonly used smoothing filtering operations, such as the average filtering, Gaussian low-pass filtering and median filtering. Yu (Yu &Chang, 2005) proposed to detect smooth regions via DCT coefficients, which was heavily dependent on high-frequency coefficients and was not robust against JPEG compression. Bayram (Bayram, Avcibas, Sankur, & Memon, 2006) employed a 188-D joint feature set composed of three types of steganalysis features to detect the smoothing operations. Their methods can achieve high detection accuracy; however, they are semi-blind and are not suitable to be used in the blind scenario. Recently, with the rapid development of computation equipment, it is preferable for the forensic task to use large dimensional feature or powerful deep learning model, such as rich model steganalysis feature with 34671-D (Qiu, Li, Luo, & Huang, 2014; Li, Luo, Qiu, & Huang, 2016) and deep learning automatically learned features (Bayar & Stamm, 2016). The rich model based feature and deep learning model really have achieved great improvements. However, the classification using large dimensional feature set and complicated learning model needs higher computation resources, larger number of images for training, longer time for training and testing, thus it is not applicable to the limited computation and storage resources (such as sensors, mobile phone). More importantly, the classification using a large dimension feature set may bear greater risk of over-fitting than that which using small dimensional feature set.

In this paper, we propose a universal smoothing filtering detector with the following goals: (1) it can simultaneously detect the commonly used smoothing filtering operations including the average filtering, Gaussian lowpass filtering and median filtering; (2) it can be robust against the commonly post operation- JPEG compression; (3) it should have a feature set with low dimension. We hope the proposed detector can satisfy the requirement of low computational resource. To this end, we firstly select high-frequency residuals elaborately to analyze the fingerprints left behind by the smoothing filtering operations, and then construct a composite feature set with small dimension for the forensic task.

2. THE PROPOSED METHOD

In this section, we first analyze the statistical differences between original images and smoothing filtered images in the high frequency residual domain. Then we employ autoregressive model and local binary patterns to extract the fingerprints left behind by the smoothing filtering operations in the residual domain. We finally introduce how to ensemble LBP and AR feature set for smoothing filtering forensics.

2.1. The High Frequency Residual

How to select an appropriate domain to analyze the fingerprints of smoothing filtering is the first critical factor for the smoothing filtering forensics. Generally, the diversity of natural image contents and post manipulations (such as the commonly used JPEG compression) always interferes with the performance of the forensic detector. The high frequency components, such as image differences or filtered residuals, contain very few image contents, so that they are always employed as being the analyzed domain to reduce the interference from image contents (Chen, Ni, & Huang, 2013; Cao, Zhao, Ni, Yu, & Tian, 2010; Kang, Stamm, Peng, & Liu, 2012, Yang, Ren, Zhu, Huang, & Shi, 2017). Inspired by our prior work (Kang, Stamm, Peng, & Liu, 2012), the median filtered residual (MFR) defined in (1) is employed as being the analyzed domain. Compared with the horizontal or vertical image differences, MFR is able to reveal the statistical information in various directions.

$$r(i,j) = \text{med}_w(x(i,j)) - x(i,j) \quad r\left(i,j\right) = med_w\left(x\left(i,j\right)\right) - x\left(i,j\right) = z(i,j) - x(i,j) \tag{1}$$

In the formula (1), $x(i,j)$, $z(i,j)$ and $r(i,j)$ are the original pixel, median filtered pixel and MFR residual pixel, respectively.

An original image, the corresponding MFR, and MFR of its 3×3average filtered version are shown in Figure 1 respectively. For the display purpose, the absolute value of MFR pixel is truncated within [0, 50], where the residual pixels (MFR=0) are shown by the black color, the residual pixels (|MFR|>= 50) are shown by the white color. From the Figure 1, it can be observed that: (1) Most of the MFR pixels as shown in Figure 1(b) and Figure 1(c) are equal or close to be 0, which indicates that MFR contains only a few image contents. (2) MFR of the average filtered image is blacker than MFR of the original image. It means that the MFR of the average filtered image has more residual pixels with small values. This phenomenon is caused by the blurring effect of the average filtering. (3) Figure 1(b) and Fib. 1(c) display obvious statistical differences in the texture and the edge region. That's because the smoothing filter makes the filtered images smoother and less noisy. For the average filtered image, the following median filtering operation in the formula (1) keeps more pixels in the texture or edge region, so MFR of the original image and MFR of its average filtered image behave statistical differences in the texture and edge regions. These properties indicate that MFR is helpful to construct an effective and robust smoothing filtering detector.

2.2. The Composite Feature

2 2.1. AR feature

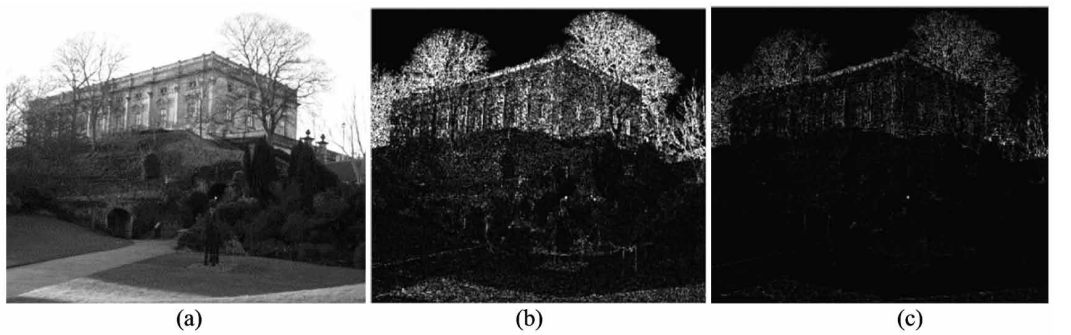
The outputs of smoothing filtering are determined by the neighborhood pixels and the filter kernel. Accordingly, it can be supposed that the smoothing filtering operation changes the relationships among neighboring pixels. To capture such relationships, we fit the MFR into an AR model, and extract AR coefficients as being a feature subset F_{AR}.

The AR coefficients are extracted from both of the vertical and horizontal directions. Extracting AR coefficients in the horizontal direction is as follows. First, concatenate all rows of MFR matrix r to generate a 1-D sequence z= [r (1,:), $r^{(LR)}$ (2,:), r (3,:), $r^{(LR)}$ (4,:), …], where

$r^{(LR)}$ (m,:) (m is the row index) is a left-right flipped version of the m^{th} row. Then, input z into an AR model as (2) to calculate AR coefficients.

$$z(t) = -\sum_{k=1}^{p} a(k)z(t-k) + \varepsilon(t) \tag{2}$$

Figure 1. An original image: (a), the corresponding MFR (b) and MFR of its 3×3 average filtered version



In the formula (2), $p, a(k), \varepsilon(t)$ represent the order, the coefficient and prediction error, respectively. Transposing the MFR matrix r, we can extract AR coefficients in the vertical direction in the same way. The average values of horizontal and vertical AR coefficients are represented as F_{AR}.

The dimension of F_{AR} is equal to the order p, which is usually small, e.g., less than 10. Hence the feature set F_{AR} is able to reveal the neighborhood relations with a low dimension. However, we also find that the AR model is not suitable for expressing other directional neighborhood relationships except the horizontal and the vertical ones. We will employ LBP to reveal neighborhood relationships in the other directions.

2.2.2. LBP-based Feature

Because of the lowpass filtering property, the smoothing filter causes obvious statistical differences between original images and smoothing images in texture regions as shown in Figure 1(b) and Figure 1(c). We suppose that some fingerprints of smoothing filtering operation may hide in the texture region. To capture the fingerprints in the texture region, we employ the LBP calculator (Ojala, Pietikainen, & Maenpaa, 2002), which is a popular useful tool to classify image textures to construct a feature set F_{LBP}. LBP has already been used in the image sharpening detection (Ding, Zhu, & Shi, 2013; Ding, Zhu, Yang, Xie, & Shi, 2014).

LBP exploits the neighborhood relationships through comparing a central MFR element r_c with its neighboring elements r_i ($i \in Z$). For l neighboring elements, the LBP calculator calculates a binary sequence $S_l = s_0 s_1 s_2 \ldots s_l$, where $s_i = f(r_c, r_i)$. If $r_i >= r_c$, $f(r_c, r_i)$ is assigned with a binary bit 1; otherwise, $f(r_c, r_i)$ is assigned with a binary bit 0. For a fixed order sequence S_l as shown in Figure 2, its decimal number D is obtained as (3).

$$D = \sum_{i=0}^{l} f(r_c, r_i) \times 2^i, where\ f(r_c, r_i) = \begin{cases} 1, & if\ r_c \le r_i \\ 0, & otherwise \end{cases} \tag{3}$$

Figure 2. The neighborhood $R_8 = r_0 r_1 \ldots r_7$ (pale green color), $R_{16} = r_8 r_9 \ldots r_{23}$ (golden color) and $R_{24} = r_{24} r_{25} \ldots r_{47}$ (gray color) using for calculating of LBP_8, LBP_{16} and LBP_{24} respectively

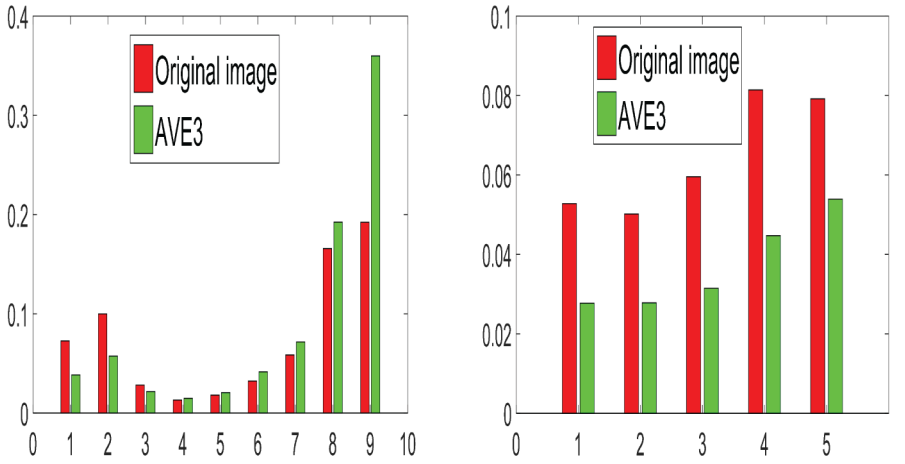
r_{33}	r_{32}	r_{31}	r_{30}	r_{29}	r_{28}	r_{27}
r_{34}	r_{14}	r_{13}	r_{12}	r_{11}	r_{10}	r_{26}
r_{35}	r_{15}	r_3	r_2	r_1	r_9	r_{25}
r_{36}	r_{16}	r_4	r_c	r_0	r_8	r_{24}
r_{37}	r_{17}	r_5	r_6	r_7	r_{23}	r_{47}
r_{38}	r_{18}	r_{19}	r_{20}	r_{21}	r_{22}	r_{46}
r_{39}	r_{40}	r_{41}	r_{42}	r_{43}	r_{44}	r_{45}

The initial LBP feature set (Ojala, Pietikainen, & Maenpaa, 2002) is not suitable to be taken directly as the feature. That's because: (1) the histogram of D is sparse; (2) the dimensions of original LBP grow exponentially with the number of neighborhood elements. For example, LBP of 3×3 neighborhood (LBP_8), 5×5 neighborhood (LBP_{16}) and 7 ×7 neighborhood (LBP_{24}) has 2^8= 256, 2^{16}= 65532 and 2^{24}= 16776192 elements respectively. Inspired by the method proposed by Ojala (Ojala, Pietikainen, & Maenpaa, 2002), we first divide LBP patterns into uniform patterns and non-uniform patterns, and then construct the feature from both two patterns. For a LBP sequence $S_l = s_0 s_1 s_2 \ldots s_l$, if the number of binary transitions (0→1 or 1→0) is no more than 2, it is taken as a uniform pattern; otherwise, it is taken as a non-uniform pattern. The following four patterns: S_l = 000 …0(all "0"), S_l = 111 …1(all "1"), S_l = 001 …0(only one "1") and S_l = 110 …1(only one "0") are typical uniform patterns.

Because the smoothing filter can be taken as an isotropic filter, we suppose that the uniform patterns containing the same number of binary bit "1" play nearly the same role when differentiating the original image from the smoothing filtered image. *To this end, the uniform patterns with the same number of binary "1" are taken as the same pattern class. This property can be used to reduce the high dimensions of the traditional LBP.* For example, "01100000" and "00000011" are taken as the same pattern class. For $LBP_{8,}$ LBP_{16}, and LBP_{24}, there are 9, 17 and 25 pattern classes respectively. The frequency of each pattern class is calculated as a feature element. Although the non-uniform patterns are very sparse, we find that they are also useful in the smoothing filtering forensics as shown in Figure 3(c). Similarly, the non-uniform patterns with the same number of binary 1 are taken as one pattern class. The frequency of each non-uniform pattern class is also calculated as a feature element. Except the 4 typical uniform patterns, there are 9-4=5, 17-4=13 and 25-4=21 non-uniform patterns in LBP_8, LBP_{16}, and LBP_{24} respectively. In the following sections, the feature set constructed by the uniform patterns and the non-uniform patterns in LBP_8, LBP_{16} and LBP_{24} are denoted by F_{8U} (9-D), F_{8NU} (5-D), F_{16U} (17-D), F_{16NU} (13-D), F_{24U} (25-D), F_{24NU} (21-D), respectively.

Figure 3 shows the statistical differentiated properties of the uniform patterns (left plot) and non-uniform patterns (right plot). For the uniform pattern feature F_{8U}, it can be seen that the frequencies of original images and3×3 averaged filtered images are significantly different at x = 1, 2, 8, 9. These indexes are corresponded to the aforementioned four typical patterns. The right plot in Figure 3 shows that all non-uniform pattern classes can also be used to differentiate the original images from the smoothing filtered images.

Figure 3. The statistical properties of F_{8U} (left) and F_{8NU} (right) on original images and 3×3 average filtered images. The x axis is the number of binary bit "1", while y axis is the average frequency of 1338 images of UCID database



Besides extracting the features from LBP_8, LBP_{16} and LBP_{24}, the relationships among 3×3 neighborhood ($R_8=r_0r_1\ldots r_7$), 5×5 neighborhood ($R_{16}=r_8r_9\ldots r_{23}$) and 7 ×7 neighborhood ($R_{24}=r_{24}r_{25}\ldots r_{47}$) are also used to construct the features. We have found that the probability of neighboring MFR elements being equal in the smoothing filtered image is higher than that in the original image. So, we construct a feature set $F_{joint}=[F_{8joint16}, F_{8joint24}, F_{16joint24}]$ from the relationships among R_8, R_{16} and R_{24}.

Before constructing the proposed feature set F_{LBP}, the whole MFR image is divided into 7×7 overlapping blocks from top left corner to the bottom right corner with increment 1 pixel. For a MFR image with resolution $M \times N$, there are totally $(M\text{-}6) \times (N\text{-}6)$ blocks. For each block, the procedure of calculating feature elements is as follows:

1. Constructing the feature sets F_{8U} and F_{8NU} in LBP_8. For the 8-bit binary sequence$S_8=s_0s_1\ldots s_7$ with k binary "1" bits ($0 \le k \le 8$), if S_8 is a uniform pattern, we add 1 to $F_{8U}(k)$; otherwise, we add 1 to $F_{8NU}(k)$. The feature sets F_{16U}, F_{16NU} in LBP_{16}, F_{24U} and F_{24NU} in LBP_{24} are constructed in the same way.
2. Constructing the feature set $F_{8joint16}$, $F_{8joint24}$, $F_{16joint24}$ based on the adjacent relationships between R_8 and R_{16}, $R8$ and R_{24}, R_{16} and R_{24} respectively. For calculating $F_{8joint16}$, we give 8 neighboring pixel groups for all elements in R_8: $r_0\{r_8\}$,$r_1\{r_9,r_{10},r_{11}\}$,$r_2\{r_{12}\}$,$r_3\{r_{13},r_{14},r_{15}\}$,$r_4\{r_{16}\}$,$r_5\{r_{17},r_{18},r_{19}\}$,$r_6\{r_{20}\}$,$r_7\{r_{21},r_{22},r_{23}\}$. For calculating $F_{8joint24}$, we also give 8 neighboring pixel groups for all elements in R_8: $r_0\{r_{24}\}$,$r_1\{r_{25},r_{26},r_{27},r_{28},r_{29}\}$,$r_2\{r_{30}\}$,$r_3\{r_{31},r_{32},r_{33},r_{34},r_{35}\}$,$r_4\{r_{36}\}$,$r_5\{r_{37},r_{38},r_{39},r_{40},r_{41}\}$,$r_6\{r_{42}\}$,$r_7\{r_{43},r_{44},r_{45},r_{46},r_{47}\}$. For calculating $F_{16joint24}$, we give16 neighboring pixel groups for all elements in R_{16}: $r_8\{r_{24}\}$,$r_9\{r_{25}\}$,$r_{10}\{r_{26},r_{27},r_{28}\}$, $r_{11}\{r_{29}\}$, $r_{12}\{r_{30}\}$, $r_{13}\{r_{31}\}$,$r_{14}\{r_{32}, r_{33},r_{34}\}$, $r_{15}\{r_{35}\}$, $r_{16}\{r_{36}\}$, $r_{17}\{r_{37}\}$, $r_{18}\{r_{38}r_{39},r_{40}\}$, $r_{19}\{r_{41}\}$, $r_{20}\{r_{42}\}$, $r_{21}\{r_{43}\}$, $r_{22}\{r_{44}, r_{45},r_{46}\}$, $r_{23}\{r_{47}\}$.Then we calculate the absolute difference between the neighboring pixels set and the residual pixel r_i. Taken the $r_1\{r_9, r_{10}, r_{11}\}$ group for example, $|r_1\text{-}r_9|$, $|r_1\text{-}r_{10}|$, $|r_1\text{-}r_{11}|$ will be calculated. If all absolute values are less than T, we will take the group $r_1\{r_9, r_{10}, r_{11}\}$as a valid element. If there are k valid elements, we will add 1 to $F_{8joint16}(k)$. The same procedure is executed for $F_{8joint24}$ and $F_{16joint24}$. We set $T=3$ in practice. For each block, the extraction procedure of F_{joint} is executed only if S_8, $S_{16}$and S_{24} are all uniform patterns.

After finishing the procedures for all $(M\text{-}6) \times (N\text{-}6)$ blocks, we unite all feature sets to get the normalized $F_{LBP=}[F_{8U}, F_{8NU}, F_{16U}, F_{16NU}, F_{24U}, F_{24NU}, F_{8joint16}, F_{8joint24}, F_{16joint24}]/(M\text{-}6) \times (N\text{-}6)$. There are 9, 5, 17, 13, 25, 21, 9, 9, and 17 elements in F_{8U}, F_{8NU}, F_{16U}, F_{16NU}, F_{24U}, F_{24NU}, $F_{8joint16}$, $F_{8joint24}$ and $F_{16joint24}$ respectively. The dimensionality of F_{LBP} is 9+5+17+13+25+21+9+17 =125. The final feature set F is a composition of F_{LBP} and F_{AR}: $F=[F_{LBP}, F_{AR}]$. In practice, we empirically set the AR order to be 10. So, the dimension of F is 125+10=135.

3. EXPERIMENTAL RESULTS

A composite database with 2676 images is used in the experiment. The composite database is composed of1338 images from the UCID database (Schaefer, &Stich, 2004) and randomly selected 1338images from the BOWS2 database (Bas, &Furon, 2006). The resolution of image is 512×512 or 512×384. All images are converted into 8-bit grayscale images at the first step. Taken the original composite database as the mother source database, three categories of image sources are created: 3×3average filtered images (AVE3), 3×3 median filtered images (MF3) and 3×3 Gaussian lowpass ($\sigma=0.5$) filtered images (GAU3). In order to test the robustness against lossy post-compression, three image sources are processed by JPEG compression with quality factor (QF = 70) and are denoted by: "AVE3 +JPEG 70," "MF3+JPEG 70," and "GAU3+JPEG 70," respectively.

We feed the proposed feature F and the GLF feature proposed by Chen (Chen, Ni, & Huang, 2013) into 3 popular machine learning tools: SVM with RBF kernel (Chang, & Lin, 2011), neural

network (NN) (Rasmus Berg Palm, 2013), and random forest (RF) (Chen, Liaw, & Breiman, 2004) to train the smoothing filtering detector. Cross-validations are used to find the best parameters for all machine learning tools. Half of original images (1338 original images) and smoothing filtered images (446 AVE3 images + 446 MF3 images + 446 GAU3 images) are used to train, the rest half is used for testing. The detection accuracy (ACC= (TPR+TPN)/2, TPR: true positive rate, TPN: true negative rate) and ROC curves are used to evaluate the performance.

At first, we test the detectors for uncompressed images. The detection accuracies are shown in Table 1 under without JPEG compression. For three kinds of classifier, our proposed detector all achieves nearly perfect performances and outperforms GLF-based detector. Especially on the RF classifier test, the proposed detector achieves about 1.8% higher detection accuracy than the GLF-based detector does. The ROC curves obtained from SVM in Figure 4(a) verify again that the proposed detector is effective for detecting smoothing filtering. For a very $FPR = 0.5\%$, the proposed detector achieves $TPR = 100\%$, which indicates that the proposed detector is effective in the harsh scenario for low FPR.

As images are usually saved as ".jpg" format, we then test the detectors' robustness against JPEG compression with QF=70. Under the JPEG compressed scenario, the JPEG 70 compressed images are taken as the negative class, while the images undergone by smooth filtering followed by JPEG 70 compression are taken as the positive class. This negative and positive pair is represented as "JPEG 70 VS smooth filtering +JPEG 70". The results in Table 1 demonstrate that our method can stably achieve ACC>97% for the JPEG compression, which indicates that it is robust against JPEG compression. On the RF classifier test, the proposed detector's accuracy is about 5.6% higher that of the GLF-based detector. In Figure 4(b), it also can be seen that the ROC curve of our proposed method is above over that of the GLF-based method. For a low FPR= 0.5%, the proposed method and the GLF-based method achieve TPR= 99.9% and TPR= 97.9% respectively. The result from uncompressed images and the result from JPEG 70 compressed images are nearly the same, which indicates that the proposed method is robust to JPEG compression. The results in Table 1 show that

Figure 4. ROC curves (obtained from SVM) showing smoothing filtering detection performance on un-compressed images (a) and JPEG 70 images (b) in the composite database

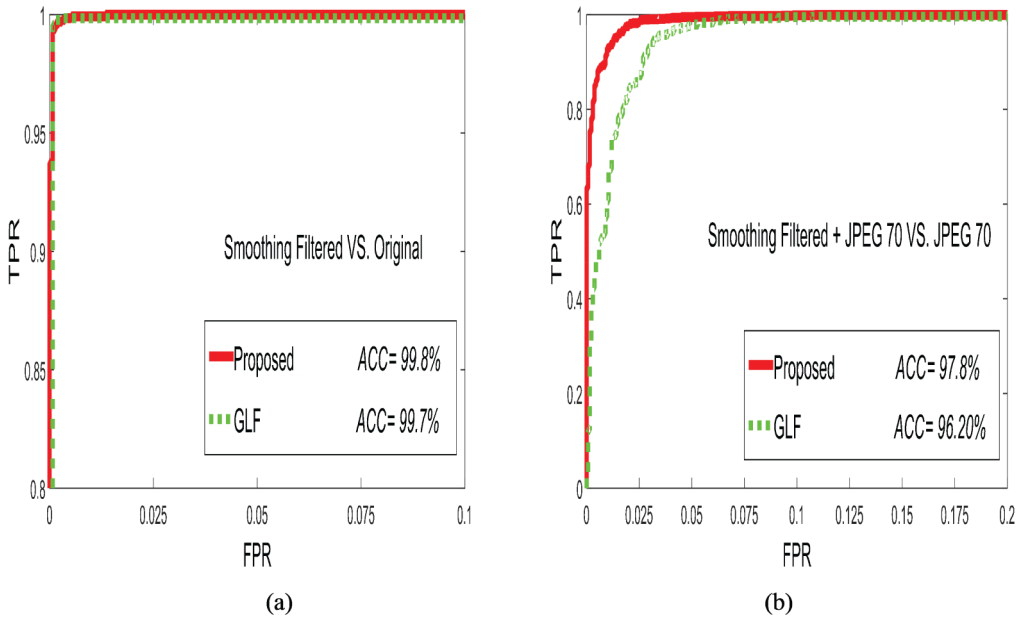


Table 1. The detection accuracy (%) of smoothing filtering detector on the composite database. "Without/QF= 70" means uncompressed and JPEG 70 compressed scenario respectively. The better results are shown in bold text

JPEG	Classifier	Proposed	GLF
Without	SVM	**99.8**	99.7
	NN	**99.6**	99.4
	RF	**99.6**	97.4
QF=70	SVM	**97.8**	96.2
	NN	**97.9**	95.1
	RF	**97.4**	91.8

the results obtained from SVM, NN, and RF are nearly the same, which indicates that the proposed method is stable and the proposed feature set probably have low risk of over-fitting because of low dimension. In the test, we find that SVM classifier achieves the best detection accuracy, while NN is the fastest.

In some applications, identifying the type of smoothing filtering is important, we model the identification as a multiple classification. The original image, its average filtered version, its median filtered version and its Gaussian filtered version are labeled as 0, 1, 2, and 3, respectively. To train a four-class classifier, we randomly select 2676/2=1338 original images, and their corresponding average filtered images, median filtered images and Gaussian filtered images as the training database. The four-class classifier is tested on the rest 2676/2=1338 original images, average filtered images, median filtered images and Gaussian filtered images. For the un-compressed scenario, the proposed detector and the GLF-based detector achieve ACC= 95.1% and ACC= 98.0% respectively. For the JPEG 70 compressed scenario, the proposed detector and the GLF-based detector achieve ACC= 92.6% and ACC= 89.5% respectively. These results reveal that the proposed detector can effectively identify the type of smoothing filtering.

We also test the generalization ability of the proposed detector. In real applications, it is probably that the training images and testing images are originated from different image sources. To test the detector's generalization ability, we first train a detector on the UCID/BOWS2 database, and then test it on BOWS2/UCID database. The detailed results in Table 2 indicate that the mismatch between trained images and tested images has a little interference with the proposed detector's accuracy. Either for the un-compressed images or JPEG 70 compressed images, the proposed detector achieves better performance than the GLF-based detector. It is worth noting that the generalization ability for the JPEG compressed scenario needs to be further improved, which is our further work.

Table 2. The generalization ability test accuracy (%) using SVM for the smoothing filtering detector. "Without/QF=70" means uncompressed and JPEG 70 compressed scenario respectively. The better results are in bold text

JPEG	Test database	Proposed	GLF
Without	UCID	**99.0**	98.8
		97.9	92.8
QF=70	BOWS2	**99.6**	98.3
		94.6	**95.5**

4. CONCLUSION

In this paper, a universal detector is developed to detect three categories of smoothing filtering: average filtering, Gaussian low-pass filtering and median filtering. In order to obtain a low dimensional, effective and robust feature set, we first analyze the statistical characteristics of smoothing operations in the median filtered residual domain, and then utilize local binary pattern and autoregressive model to construct the forensic features. We develop a method to reduce the dimensions of LBP-based features, which can reduce the exponentially growing dimensions to the linearly growing dimensions. Experimental results verify that the proposed detector is an effective and robust detector, and it outperforms previous method. In the future, we will further improve the smoothing filtering detector's robustness against JPEG compression.

ACKNOWLEDGMENT

This work was partially supported by NSFC (Grant Nos. 61702429, U1536204, 61772571, and 61379155), the Scientific Research of Sichuan Provincial Education Department (Grant No. 17ZB0450,17ZA0407), Doctoral Research Fund of Southwest University of Science and Technology (Grant No. 16zx7104) and the Fund of Fundamental Sichuan Civil-military Integration Institute.

REFERENCES

Barni, M., Cancelli, G., & Esposito, A. (2010). Forensics aided steganalysis of heterogeneous images. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 1690–1693).

Bas, P., & Furon, T. (2006). *Break Our Watermarking System* (2nd ed.). Retrieved from http://bows2.ec-lille.fr/

Bayar, B., & Stamm, M. (2016). A Deep learning approach to universal image manipulation detection using a new convolutional layer. In *Proc. of the 4th ACM Workshop on Information Hiding and Multimedia Security* (pp. 5-10). doi:10.1145/2909827.2930786

Bayram, S., Avcibas, I., Sankur, B., & Memon, N. (2006). Image manipulation detection. *Journal of Electronic Imaging*, *15*(4), 17–30. doi:10.1117/1.2401138

Cao, G., Zhao, Y., Ni, R. R., Yu, L. F., & Tian, H. W. (2010). Forensic detection of median filtering in digital images. In *Proc. Of 2010 IEEE International Conference on Multi-media and Expo* (pp. 89-94). doi:10.1109/ICME.2010.5583869

Chang, C. C., & Lin, C. J. (2011). LIBSVM: A library for support vector machines. [TIST]. *ACM Transactions on Intelligent Systems and Technology*, *2*(3), 27.

Chen, C. L., Ni, J. Q., & Huang, J. W. (2013). Blind Detection of Median Filtering in Digital Images: A Difference Domain Based Approach. *IEEE Transactions on Image Processing*, *22*(12), 4699–4710. doi:10.1109/TIP.2013.2277814 PMID:23955753

Chen, C., Liaw, A., & Breiman, L. (2004). Using random forest to learn imbalanced data. Retrieved from http://www.stat.berkeley.edu/~breiman/RandomForests/cc_software.htm

Ding, F., Zhu, G. P., & Shi, Y. Q. (2013). A novel method for detecting image sharpening based on local binary pattern. In *Proc. 12th Int. Workshop on Digital Forensics and Watermarking*.

Ding, F., Zhu, G. P., Yang, J., Xie, J., & Shi, Y. Q. (2014). Edge perpendicular binary coding for USM sharpening detection. *IEEE Signal Processing Letters*, *22*(3), 327–331. doi:10.1109/LSP.2014.2359033

Dun-Yu, H., & Soo-Chang, P. (2005). Detecting digital tampering by blur estimation. In *Proc. of the First International Workshop on Systematic Approaches to Digital Forensic Engineering* (pp. 264–278). doi:10.1109/SADFE.2005.8

Kang, X. G., Stamm, M. C., Peng, A. J., & Liu, K. J. R. (2012). Robust median filtering forensics based on the autoregressive model of median filter residual. In *Proc. of APSIPA annual submit conference* (pp. 53-66).

Kodovský, J., & Fridrich, J. (2014). Effect of image down-sampling on steganographic security. *IEEE Transactions on Information Forensics and Security*, *9*(5), 752–762. doi:10.1109/TIFS.2014.2309054

Li, H. D., Luo, W. Q., Qiu, X. Q., & Huang, J. W. (2016). (accepted). Identification of various image operations using residual-based features. *IEEE Transactions on Circuits and Systems for Video Technology*.

Niu, Y. K., Zhao, Y., & Ni, R. R. (2017). Robust median filtering detection based on local difference descriptor. *Signal Processing Image Communication*, *53*, 65–72. doi:10.1016/j.image.2017.01.008

Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). Multiresolution gray-scale and rotation in-variant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence*, *24*(7), 971–987. doi:10.1109/TPAMI.2002.1017623

Pevný, T., Bas, P., & Fridrich, J. (2010). Steganalysis by subtractive pixel adjacency matrix. *IEEE Transactions on Information Forensics and Security*, *5*(2), 215–224. doi:10.1109/TIFS.2010.2045842

Qiu, X. Q., Li, H. D., Luo, W. Q., & Huang, J. W. (2014). A universal image forensic strategy based on steganalytic model. In *Proc. of the 2nd ACM workshop on Information hiding and multimedia security* (pp. 165-170). doi:10.1145/2600918.2600941

Rasmus Berg Palm. (2013). Neural network codes. Retrieved from https://github.com/rasmusbergpalm/DeepLearnToolbox/tree/master/NN

Schaefer, G., & Stich, M. (2004). UCID-An uncompressed color image database. In *Proc. of SPIE, Storage and Retrieval Methods and Applications for Multimedia* (pp. 472–480).

Yang, J., Ren, H., Zhu, G., Huang, J., & Shi, Y. Q. (2017). Detecting median filtering via two-dimensional AR models of multiple filtered residuals. *Multimedia Tools and Applications*.

Yuan, H. D. (2011). Blind Forensics of Median Filtering in Digital Images. *IEEE Transactions on Information Forensics and Security*, *6*(4), 1335–1345. doi:10.1109/TIFS.2011.2161761

Zhang, Y. J., Li, S. G., Wang, S. L., & Shi, Y. Q. (2014). Revealing the traces of median filtering using high-order local ternary patterns. *IEEE Signal Processing Letters*, *21*(3), 275–280. doi:10.1109/LSP.2013.2295858

Anjie, Peng received the BS, MA and PhD degree from Sun Yat-sen University in 2005, 2007, and 2015 respectively. He is currently working in the School of Computer Science and Technology, Southwest University of Science and Technology, Mianyang, China. His research interests include multimedia forensics and machine learning.

Gao,Yu is currently working toward the MS degree at the School of Computer Science and Technology, Southwest University of Science and Technology, Mianyang, China. His research interests include multimedia forensics and machine learning.

Yadong Wu is a Professor with the College of Computer Science and Technology, Southwest University of Science and Technology. He received his PhD degree from University of Electronic Science and Technology of China in 2006. His research interest includes visualization and visual analytics, virtual reality and human computer interaction, and computer vision.

Qiong Zhang obtained his PhD degree in 2016 and started to work in Sun Yat-sen University as a research scientist in early 2017. Before that, he did research in McGill University as a visiting researcher from 2015 to 2016. Dr. Zhang has a diverse background both in academic research and technology innovations. He also holds professional certificates in technology entrepreneurship from University of Pennsylvania and University of Maryland.

Xiangui Kang received the BS degree from Peking University, Beijing, China, and the PhD degree from Sun Yat-Sen University, Guangzhou, China. He was a visiting scholar with University of British-Columbia in 2014, with University of Maryland-College Park during 2011–2012, and with New Jersey Institute of Technology during 2004–2005. He is currently a professor with the School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, China. His research interests include information forensics, watermarking, and multimedia communications and security. Dr. Kang is a member of IMV Technical Committee of Asian-Pacific Signal and Information Processing Association and was a member of IEEE ComSoc's Multimedia Communications Technical Committee. He has published more than 70 research papers in major international academic journals and peer-reviewed conference proceedings. He is an area editor for Signal processing: Image Communication and EURASIP Journal on Image and Video Processing.