Computers, Materials & Continua CMC, vol.65, no.3, pp.2217-2231, 2020

Multi-Purpose Forensics of Image Manipulations Using Residual-Based Feature

Anjie Peng[1], Kang Deng[1], Shenghai Luo[1] and Hui Zeng[1, 2, *]

Abstract: The multi-purpose forensics is an important tool for forge image detection. In this paper, we propose a universal feature set for the multi-purpose forensics which is capable of simultaneously identifying several typical image manipulations, including spatial low-pass Gaussian blurring, median filtering, re-sampling, and JPEG compression. To eliminate the influences caused by diverse image contents on the effectiveness and robustness of the feature, a residual group which contains several high-pass filtered residuals is introduced. The partial correlation coefficient is exploited from the residual group to purely measure neighborhood correlations in a linear way. Besides that, we also combine autoregressive coefficient and transition probability to form the proposed composite feature which is used to measure how manipulations change the neighborhood relationships in both linear and non-linear way. After a series of dimension reductions, the proposed feature set can accelerate the training and testing for the multi-purpose forensics. The proposed feature set is then fed into a multi-classifier to train a multi-purpose detector. Experimental results show that the proposed detector can identify several typical image manipulations, and is superior to the complicated deep CNN-based methods in terms of detection accuracy and time efficiency for JPEG compressed image with low resolution.

Keywords: Digital image forensics, partial correlation, auto-regression, multi-classification.

1 Introduction

As an arsenal for digital image authentication, the forensics of image manipulations which can unveil the forgeries in an image effectively, has become a very important task in digital image forensics [Wang and Zhang (2020)]. Most of existing methods are targeted forensic algorithms, which aim at detecting commonly used manipulations, such as median filtering [Wang and Zhang (2020); Luo, Peng, Zeng et al. (2019); Yang, Ren, Zhu et al. (2018); Chen, Kang, Liu et al. (2015)], re-sampling [Chen, Ni, Shen et al. (2017)], compression [Luo, Huang and Qiu (2010)], contrast enhancement [Stamm and Liu (2010)], histogram equalization [Mauro, Ehsan and Benedetta (2018)], sharpening

[1] Southwest University of Science and Technology, Mianyang, 621010, China.

[2] Binghamton University, State University of New York, NewYork, 13902, USA.

[*] Corresponding Author: Hui Zeng. Email: zengh5@mail2.sysu.edu.cn.

Received: 14 April 2020; Accepted: 12 June 2020.

CMC. doi:10.32604/cmc.2020.011006 www.techscience.com/journal/cmc

[Cao, Zhao and Ni (2011)], de-blurring [Zhang, Xiao, Xue et al. (2019)], etc., However, in practice, those targeted detectors may suffer from some drawbacks. Firstly, as the prior knowledge about the investigated image is usually unknown, we do not know which targeted detector to choose for prediction. An alternative way is running multiple tests using various targeted binary detectors. However, as each targeted detector employs a special feature, it needs to gather all kinds of different feature sets from the test image which is so tedious and time consuming. Therefore, developing a multi-purpose forensic method using a universal feature set that can identify various image operations simultaneously is of great importance.

Several multi-purpose forensics have been proposed via hand-crafted features [Fan, Wang and Cayre (2015); Jeong, Moon and Eom (2015); Li, Luo, Qiu et al. (2018)] and self-learning features [Bayar and Stamm (2018); Chen, Kang, Shi et al. (2019); Liu, Guan, Zhao et al. (2018)]. One typical hand-crafted feature is proposed by Li et al. [Li, Luo, Qiu et al. (2018)]. Supposing that image manipulations modify many pixels and the degree of modification is much greater than steganography, they employed the tools from steganalysis, such as spatial-domain rich model (SRM), local binary pattern (LBP), to perform multi-purpose forensics, and achieved excellent performance for identifying various image manipulations. Benefiting from the powerful computing ability of GPU, deep CNN models are employed to automatically learn image manipulation traces [Bayar and Stamm (2018); Chen, Kang, Shi et al. (2019); Liu, Guan, Zhao et al. (2018)]. Bayar et al. [Baya and Stamm (2018)] designed a constrained CNN architecture called MISLnet with 5 convolution layers and 3 fully connected layers to adaptively learn a content-independent high-level features, and accurately identified the type of manipulation for un-compressed images. Chen et al. [Chen, Kang, Shi et al. (2019)] proposed a densely connected CNN called Dense-CNN with 3 dense blocks containing 8 convolution layers and 1 fully connected layer for multi-purpose forensics, and obtained better performances than Li's method [Li, Luo, Qiu et al. (2018)] and Liu's CNN [Liu, Guan, Zhao et al. (2018)] in terms of average classification accuracies under JPEG compression. However, the issues about the time efficiency and the robustness against JPEG compression are still needed to be addressed. Unfortunately, the huge dimensional SRM model and complicated deep CNN model need longer time for feature extracting and training, or higher computation resources, thus are very time-consuming.

In this paper, we consider the identification of 4 typical image manipulations including spatial low-pass Gaussian blurring (GB), median filtering (MF), re-sampling (RS) and JPEG compression (JP), and try to propose a hand-crafted feature set to improve the time efficiency and robustness against JPEG compression for multi-purpose forensics. Firstly, we employ a diverse residual group to enhance the weak traces of image manipulations left in the JPEG compressed image. Then, considering that multi-purpose forensics is more difficult than the target forensics, besides of the popular autoregressive coefficient [Kang, Stamm, Peng et al. (2013); Yang, Ren, Zhu et al. (2018)] and transition probability [Li, Luo, Qiu et al. (2018); Pevný, Bas and Fridrich (2010)], we first propose ***partial correlation coefficients*** as a feature set to measure the correlations among neighboring pixels changed by multiple manipulations as fully as possible. It is worth mentioned that, based on the link between the partial correlation coefficient and the autoregressive coefficient, we can extract them simultaneously via an autoregressive model. Therefore, the proposed method does not increase any time in the feature extraction compared with our previous work which only extracting autoregressive coefficients [Kang, Stamm, Peng et al. (2013)]. After dimension reductions, a feature set with small dimension is obtained, which allows us to accelerate the training and testing process. Experimental results show that the proposed detector performs better than the complicated deep CNN-based methods [Bayar and Stamm (2018); Chen, Kang, Shi et al. (2019)] for JPEG compressed images with low resolutions.

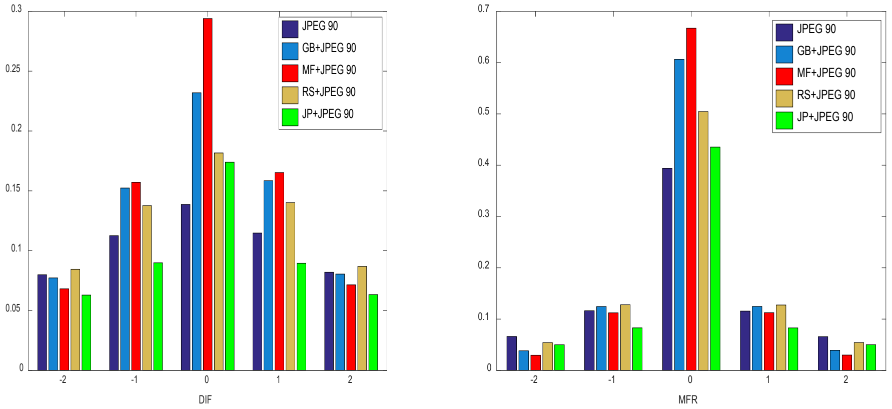

Figure 1: Average histograms of DIF (left) and MFR (right) estimated from unaltered images and corresponding manipulated versions. All images are JPEG 90 post-compressed (denoted by "+")

2 Proposed method

2.1 Residual group

It is well known that image manipulations will modify the pixels in the original image and thus inevitably change the intrinsic statistics. If these statistical changes are directly captured from image pixel domain, the effectiveness and robustness of the forensic algorithm will be disturbed by diverse natural image contents. Therefore, as many targeted forensic works [Luo, Peng, Zeng et al. (2019); Yang, Ren, Zhu et al. (2018); Kang, Stamm, Peng et al. (2013); Chen, Ni, Shen et al. (2017); Chen, Shi and Su (2009)], we first introduce residuals from image pixel domain and then extract feature from the residual domain. Considering that the proposed scheme is multi-purpose, we collect the residuals from some targeted algorithms, including median filtering residual (MFR) and its difference (MFRD) used for median filtering forensics [Kang, Stamm, Peng et al. (2013); Peng and Kang (2016)], the difference (DIF) for re-sampling and JPEG compression forensics [Chen, Ni, Shen et al. (2017); Chen, Shi and Su (2009)], and package them as a residual group. Specially, the residual group has 17 residuals containing 8 DIFs, 1 MFR and 8 MFRDs. The definitions of the 1st-order DIF, MFR and the corresponding MFRD for an image X are given in Eq. (1), where the superscript (h, v) $\in \{(0, 1), (0, -1), (1, 0), (-1, 0), (1, -1), (-1, 1), (1, 1), (-1, -1)\}$ denotes the direction of residual, $\mathrm{MF}_3\{\}$ is a 2-D median filtering operation with 3×3 filtering window.

To preliminarily show the differentiated ability of residuals, we draw average histograms of DIF $^{(1,\,0)}$ (X) and MFR(X) for the unaltered, GB, MF, RS and JP images of size 128×128. The histograms in Fig. 1 are evaluated from 1000 randomly selected images of Bossraw [Bas, Filler and Pevný (2011)]. Please refer to Section 3 for more details of manipulation parameters. For clear observations, the ranges of residuals are limited to [-2, 2]. Under JPEG 90 compression, the histograms of DIF(X) and MFR(X) behave different distributions for different manipulations. The similar results can also be drawn from MFRD. Based on these observations, it is expected that a diverse and effective feature set can be generated from the residual group.

$$\mathrm{DIF}^{(h,\,v)}\,(X\,(i,j))=X\,(i,j)\text{-}X\,(i+h,j+v)$$
$$\mathrm{MFR}\,(X\,(i,j))=\mathrm{MF}_3\{X\,(i,j)\}\text{-}X\,(i,j)$$
$$\mathrm{MFRD}^{(h,\,v)}\,(X\,(i,j))=\mathrm{MFR}\,(X\,(i,j))\text{-}\mathrm{MFR}\,(X\,(i+h,j+v)) \qquad (1)$$

2.2 Feature extraction using partial autocorrelation and autoregressive model

In general, different manipulations modify the unaltered image in different ways, which means that they may disturb relationships of neighboring pixels with different degrees. Correlation is such an important neighborhood relationship in forensics, steganalysis and watermarking [Yang, Ren, Zhu et al. (2018); Kang, Stamm, Peng et al. (2013); Chen, Ni, Shen et al. (2017); Kang, Liu, Yang et al. (2019); Peng, Lin, Zhang et al. (2019)]. We employ partial autocorrelation coefficient (PAC) and autoregressive coefficients (ARC) to measure the neighborhood correlations disturbed by image manipulations. Noticing that, the PAC purely measures the correlation degree of neighboring pixels, and the ARC measures the linear dependence magnitude of neighboring pixels. Hence, it is expected that the combination of PAC and ARC performs better than single ARC in the multi-purpose forensics.

The PAC is a pure version of commonly used autocorrelation coefficient. Given a signal sequence Z_t, PAC measures the correlation between a signal and a delayed copy of itself, after excluding the effect of other delayed versions. Specially, the p-order PAC φ_{pp} defined in (2) measures the correlation between Z_t and Z_{t-p} with the dependency of Z_{t-1} through to Z_{t-p+1} being removed. In Eq. (2), E(.) represents an expectation function. Employing PAC has two advantages: (1) PAC feature have very small dimensions, because PAC of high order (high order means large distance between two pixels) are very weak [Pevný, Bas and Fridrich (2010)]. (2) PAC and autoregressive coefficients ARC can be simultaneously extracted from an autoregressive model. In the following, we will introduce how to extract PAC and ARC simultaneously.

$$\varphi_{pp}=\frac{E[(z_t-\hat{E}(z_t))(z_{t-p}-\hat{E}(z_{t-p}))]}{\sqrt{E(z_t-\hat{E}(z_t))^2}\sqrt{E(z_{t-p}-\hat{E}(z_{t-p}))^2}} \qquad (2)$$

$$\textit{where } \hat{E}(z_t)=E(z_t|z_{t\text{-}1},\,.\,.\,.\,,z_{t-p+1})$$

$$
\begin{aligned}
&z_t = \boxed{\varphi_{11}} z_{t-1} + \varepsilon_{t1} \\
&z_t = \varphi_{21} z_{t-1} + \boxed{\varphi_{22}} z_{t-2} + \varepsilon_{t2} \\
&z_t = \varphi_{31} z_{t-1} + \varphi_{32} z_{t-2} + \boxed{\varphi_{33}} z_{t-3} + \varepsilon_{t3} \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
&z_t = \enclose{circle}{\varphi_{p1}} z_{t-1} + \enclose{circle}{\varphi_{p2}} z_{t-2} + \enclose{circle}{\varphi_{p3}} z_{t-3} + \ldots + \enclose{circle}{\boxed{\varphi_{pp}}} z_{t-p} + \varepsilon_{tp}
\end{aligned}
$$

Figure 2: The ARC depicted by blue circle and PAC depicted by red rectangle

From Eq. (2), it can be inferred that the PAC φ_{pp} is the p^{th} coefficient of a *p*-order autoregressive (AR) model [Steven (1988)]. We show the relationships between PAC and ARC in Fig. 2. When using Burg method [Steven (1988)] to recursively estimate the coefficients of a *p*-order AR model, the coefficients of 1-order, …, (*p*-1)-order (i.e., $\varphi_{11}, \varphi_{21}, \varphi_{22}, \varphi_{31}, \ldots, \varphi_{p-1,p-1}$ in Fig. 2) should be calculated firstly. So, the PAC coefficients $\varphi_{11}, \varphi_{21}, \varphi_{22}, \varphi_{31}, \ldots, \varphi_{p-1,p-1}$ can be obtained in the process of calculating *p*-order AR coefficients. That is, we can estimate the ARC and the PAC simultaneously. Compared with our previous work which only extract ARC [Kang, Stamm, Peng et al. (2013)], we can simultaneously get ARC and PAC in this work but without increasing the feature extracting time. In the next paragraph, we will introduce how to extract ARC and PAC from a residual matrix.

Because the AR model is used for 1-D signal, it is needed to transform a 2-D residual matrix into a 1-D vector Z_t firstly. In order to capture statistical changes in different directions, the transformation is executed along with the row and column directions respectively to obtain $Z_t^{(r)}$ and $Z_t^{(c)}$. Then, estimating the ARC and PAC coefficients from $Z_t^{(r)}$ and $Z_t^{(c)}$ respectively, and averaging them to obtain a (2*p*+1)-dimension feature which is composed by *p* ARCs, *p* PACs and 1 prediction error. To supply enough statistical samples for small image blocks, a new concatenated vector $[Z_t^{(r)}, Z_t^{(c)}]$ is also used to extract the feature. Similarly, we will get another (2*p*+1)-dimension feature. Through composing all feature elements, we get a feature set with 2×(2*p*+1) dimension from each residual.

If directly composing all feature subsets from totally 17 residuals, it will get a composite feature with 17×2×(2*p*+1) dimensions. Assuming the positive-negative equality and rotational symmetry property of DIF, we reduce the dimensions of the feature extracted by DIF from 8×2×(2*p*+1) to 2×2×(2*p*+1). Specially, only 4 DIFs out of all 8 DIFs including DIF $^{(1,0)}$ (*X*), DIF $^{(0,1)}$ (*X*), DIF $^{(1,-1)}$ (*X*) and DIF $^{(1,1)}$ (*X*) are used to extract the feature. The feature extracted from DIF $^{(1,0)}$ (*X*) and DIF $^{(0,1)}$ (*X*) are averaged to form the first part, and the feature extracted from DIF $^{(1,1)}$ (*X*) and the DIF $^{(1,-1)}$ (*X*) are averaged to form the second part. After concatenation, we get a 2×2(2*p*+1) dimensional feature set extracted from DIF. Similarly, we get another 2×2(2*p*+1) dimensional feature set extracted from MFRD in the same way.

In summary, the proposed feature set based on ARC and PAC ($\mathrm{F_{ARCPAC}}$) is extracted from DIF, MFR and MFRD respectively, which is calculated as Eq. (3). For a *p*-order AR model, $\mathrm{F_{ARCPAC}}$ has 5×2(2*p*+1) dimensions. Because of the truncation property of PAC [Steven (1988)], we use the PAC distributions as shown in Fig. 3 to estimate the order of

AR model, where the same 1000 JPEG 90 compressed images used in Fig. 1 is used. Fig. 3 shows that the 10th-order PAC is nearly to be 0. Therefore, we empirically set the order p as 10, and get a 210-D feature F_{ARCPAC}. From Fig. 3, it also can be seen that 10 PACs are able to distinguish 5 types of images, especially the first 5 PACs.

F_{ARC}= [$F_{ARC\text{-}MFR}$, $F_{ARC\text{-}DIF}$, $F_{ARC\text{-}MFRD}$]

F_{PAC}= [$F_{PAC\text{-}MFR}$, $F_{PAC\text{-}DIF}$, $F_{PAC\text{-}MFRD}$]

F_{ARCPAC}= [F_{ARC}, F_{PAC}] (3)

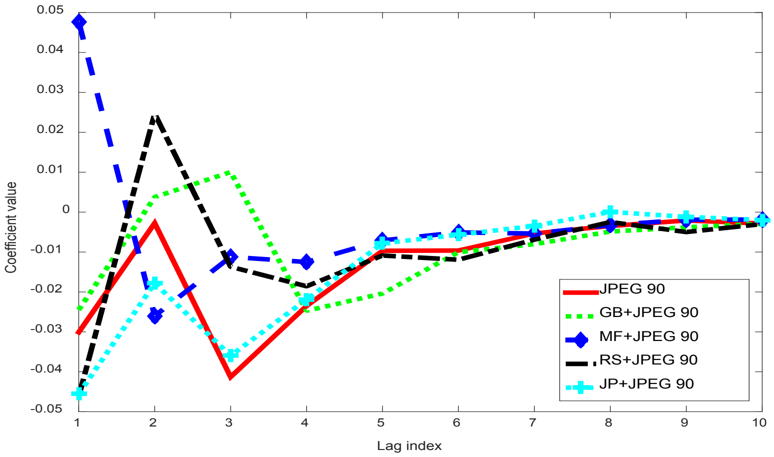

Figure 3: Illustrations of the average PAC estimated from MFRD of 1000 JPEG 90 compressed images corresponding different manipulated versions

2.3 Feature extraction using transition probability of Markov chain

The transition probability (TP) of Markov chain, which expresses the conversion relationships among adjacent pixels, is an effective non-linear measurement of neighborhood relationships in the forensics [Chen, Shi and Su (2009); Li, Luo, Qiu et al. (2018)]. To make up for the short board of F_{ARPAC} whose detection accuracy for JPEG compression is not high, the TP of the nth-order Markov chain is also employed to construct the feature.

The TP extracted from MFR, DIF, MFRD are denoted by $F_{TP\text{-}MFR}$, $F_{TP\text{-}DIF}$ and $F_{TP\text{-}MFRD}$, respectively. Before calculating TP, the residual is truncated into [-T, T]. F_{TPMFR} is calculated along with 8 directions to get 8 TP matrices as SPAM [Pevný, Bas and Fridrich (2010)]. As for $F_{TP\text{-}DIF}$ and $F_{TP\text{-}MFRD}$, we only extract the TP whose direction is in accord with the direction of the residual, and get 8 TP matrices for each. Taken the residual DIF $^{(1,\ 0)}$ (X) for example, we only calculate the TP along with horizontal left-right direction. Each TP matrix has $(2T+1)^n$ elements. Based on symmetric property of residual as shown in Fig. 1 and rotation invariance of TP [Pevný, Bas and Fridrich (2010)], we reduce the dimension of TP as Eq. (4), where $\alpha_k \in \{-T,...,T\}, 1 \leq k \leq n$, which makes the dimension of the reduced version $S_{low}^{(h,v)}$ is about 1/4 of that of the original TP as in Eq. (5). We experimentally set T=1 and n=4 to make a balance between the detection accuracy and the feature dimensionality. Under these settings, the dimension of original TP is $(2+1)^4=81$, while the dimension of $S_{low}^{(h,v)}$ is only 25. After dimension reduction for all TPs of each residual, 4 TP matrices along with horizontal and vertical directions are averaged as the first part, and the other 4 matrices in diagonal and mirror-diagonal directions are averaged as the second part. After concatenating the first and second part, we will get 50-D $F_{TP\text{-}DIF}$, 50-D $F_{TP\text{-}MFR}$ and 50-D $F_{TP\text{-}MFRD}$, and obtain a composite feature 150-D F_{TP} finally.

$$S_{low}^{(h,v)}(\alpha_1,\ldots,\alpha_n)=\frac{(S^{(h,v)}(\alpha_1,\ldots,\alpha_n)+S^{(h,v)}(-\alpha_1,\ldots,-\alpha_n)+S^{(h,v)}(\alpha_n,\ldots,\alpha_1)+S^{(h,v)}(-\alpha_n,\ldots,-\alpha_1))}{4} \tag{4}$$

$$|\,S_{low}^{(h,v)}\,|=\begin{cases}\dfrac{(2T+1)^n+(2T+1)^{\frac{n+1}{2}}+(2T+1)^{\frac{n-1}{2}}+1}{4}, & if\ n\ is\ odd\\[2ex] \dfrac{(2T+1)^n+2(2T+1)^{\frac{n}{2}}+1}{4}, & if\ n\ is\ even\end{cases} \tag{5}$$

2.4 Summary of the proposed method

The proposed feature set $F_{ARCPACTP}$ with 360 dimensions is formed by incorporating 100-D F_{ARC}, 110-D F_{PAC} and 150-D F_{TP} as in Eq. (6). The procedure of extracting $F_{ARCPACTP}$ from an image X is summarized as follows:

(1) Getting DIF, MFR and MFRD respectively as Eq. (1);

(2) Calculating F_{ARC} and F_{PAC}, then concatenating them to form F_{ARCPAC};

(3) Calculating $F_{TP\text{-}MFR}$, $F_{TP\text{-}DIF}$ and $F_{TP\text{-}MFRD}$, then concatenating them to form F_{TP};

(4) Concatenating F_{ARC}, F_{PAC} and F_{TP} to form the proposed $F_{ARCPACTP}$.

$$F_{ARCPACTP}=[F_{ARC},\ F_{PAC},\ F_{TP}] \tag{6}$$

We frame the multi-purpose forensics as a multiclass classification to identify the type of image manipulation. In this work, we consider the identification of 4 commonly used manipulations including GB, MF, RS and JP from original image, thus a 5-class classification problem is framed. As for the supervised learning, one-versus-one, one-versus-rest, and undirected cyclic graph are commonly used schemes to solve the multi-classification [Mendialdua, Echegaray, Rodriguez et al. (2016)]. The drawback of one-versus-one scheme is that it bears the burden having more binary classifiers. However, it may obtain higher detection accuracy in the multi-classification [Li, Luo, Qiu et al. (2018)]. For a limited number of image manipulations, the complexity of multi-classification based on one-versus-one scheme is acceptable. Thus, we employ the one-versus-one scheme in the proposed method. For the 5-class classification problem in this work, $C_5^2=10$ bi-classifiers are trained from 5 classes of training samples. In the testing, a test image will be predicted by 10 bi-classifiers, and the result is obtained by majority voting among 10 predicted labels.

3 Experimental results

3.1 Experiment setting

In order to evaluate the effectiveness of the proposed method, the raw versions of widely used BOSSbase 1.01 containing 10000 images called as Bossraw [Bas, Filler and Pevný (2011)] and Raise containing 8156 images [Dang-Nguyen, Pasquini, Conotter et al. (2015)] called as Raiseraw are selected as mother databases. All raw images are converted to 8-bit gray images by the Adobe Photoshop cc 2018 before further processing. To show the performance of small image block, we centrally crop 4 non-overlapped blocks of size 128×128, 32×32 and 16×16 from the original image, and randomly select 30000 unaltered images for each resolution from each database. For each unaltered image, we create 4 manipulated counterparts with randomly selected parameter in Tab. 1. As a result, we get totally 30000×5=150000 images for each size from each database. In order to test the robustness against JPEG compression, each unaltered image and its 4 manipulated versions are JPEG compressed with QF in {90, 80}. In summary, with setting 2 source databases, 5 types of image operation, 2 kinds of JPEG compression, and 3 image sizes, we totally obtain 2×5×2×3=60 kinds of databases, where each database has 30000 images.

Table 1: Types of image manipulations and their parameters used in the experiments

Manipulation	Parameters
Median filtering (MF)	window size: 3×3, 5×5
Gaussian blurring (GB)	window size: 3×3
	σ : 0.7, 0.8,0.9,1.0,1.1,1.2
Resampling (RS)	interpolation method: bicubic
	scaling factor: 0.8, 1.1, 1.4, 1.7, 2.0
JPEG Compression (JP)	quality factor (QF): 50, 55, 60, 65, 70, 75

SVM with RBF kernel is chosen as the bi-classifier for the proposed method. A randomly selected half image in database is used for training, and the remaining half is used for testing. A five-fold cross validation is performed in the training to search the optimal hyper-parameters. The state-of-the-arts, MISLnet [Bayar and Stamm (2018)] and Dense-CNN [Chen, Kang, Shi et al. (2019)] running on Caffe are used for comparisons. To give more training samples for deep CNN methods, 5/6 image of each kind are used for training and the rest 1/6 are for testing, i.e., 5×25000=125000 training samples, 5×5000=25000 testing samples. Considering that the architecture of MISLnet is not suitable for image of size 16×16, we only report its result for size of 32×32 and 128×128. The detection accuracy (*Acc*) is used to evaluate detector's performance. Hereafter, the same experimental settings is adopted unless specially mentioned.

$$Acc=\frac{\#correctly\ predicted\ samples}{\#total\ testing\ samples} \tag{7}$$

3.2 Performance evaluation of feature subsets

In this sub-section, we first empirically demonstrate the results of F_{TP} with different settings of Markov chain order *n* and truncated threshold *T*. As the proposed multi-classifier adopt one-versus-one strategy, its overall performance is mainly depended on the performance of each bi-classifier. Therefore, we select the parameter based on the *Acc* of F_{TP} on 10 bi-classifiers. Because the large *T* and *n* will cause huge dimension of F_{TP}, we only compare the proposed F_{TP}(*n*=4, *T*=1, 150-D) with the typical settings after dimension reduction F_{TP}(*n*=3,*T*=3, 600-D) and F_{TP}(*n*=4,*T*=2,1014-D). Tab. 2 lists the detailed results on 10 binary classifiers for 128×128 JPEG 80 compressed images from Bossraw. It is observed that the proposed F_{TP}(*n*=4, *T*=1, 150-D) has the least feature dimension (about 1/4, 1/6 of other two), but performs only a little worse than other two, even performs best among 3 kinds of F_{TP} on the "Unaltered VS RS" test. Hence, we select the proposed setting F_{TP} (*n*=4, *T*=1, 150-D), and denote it by F_{TP} for short in the following.

Table 2: *Acc* (%) of feature on 10 SVM bi-classifiers for 128×128 JPEG 80 compressed images from Bossraw database. Best results of each column are displayed in bold

Feature set	Unaltered "VS"				MF "VS"			GB "VS"		RS "VS"
	MF	GB	RS	JP	GB	RS	JP	RS	JP	JP
F_{TP} (*n*=3, *T*=3,600-D)	95.8	94.7	83.9	**95.8**	91.1	96.7	**99.5**	93.1	99.4	97.4
F_{TP} (*n*=4, *T*=2,1014D)	96.2	94.9	83.7	95.4	90.9	96.9	**99.5**	93.0	**99.5**	97.3
F_{TP} (*n*=4, *T*=1, 150-D)	96.0	94.5	84.5	94.8	90.2	96.1	99.4	92.3	99.0	96.9
F_{PAC}	95.3	95.7	88.6	78.3	93.1	96.5	97.0	95.4	97.9	91.0
F_{ARC}	95.1	95.8	88.3	73.4	93.2	96.5	96.7	95.3	97.5	90.3
F_{TP}+ F_{PAC}	96.6	96.3	89.0	95.1	93.4	97.3	**99.5**	95.8	**99.5**	**97.7**
F_{TP}+ F_{ARC}	96.6	**96.4**	89.0	95.1	**93.7**	**97.4**	**99.5**	95.9	**99.5**	**97.7**
F_{PAC} +F_{ARC}	95.3	95.8	88.9	78.4	93.2	96.6	97.0	95.5	98.0	91.2
$F_{ARCPACTP}$	**96.7**	**96.4**	**89.7**	95.1	**93.7**	**97.4**	**99.5**	**96.0**	**99.5**	**97.7**

Table 3: Confusion matrix (%) of identifying manipulations for 32×32 image. The first, second and third results in cell are for the proposed method, Dense-CNN and MISLnet respectively. The asterisks "*" denote that the corresponding values are below 1%. The best identification rate is in bold text

QF	Actual\ Predicted	Database (image size: 32×32)									
		Bossraw					Raiseraw				
		Unaltered	MF	GB	RS	JP	Unaltered	MF	GB	RS	JP
90	Unaltered	79.7	5.6	3.7	9.6	1.4	79.0	5.5	3.5	10.5	1.5
		80.3	4.2	3.2	12.0	*	**80.7**	2.6	1.6	14.5	*
		69.3	8.1	3.5	9.1	10.0	65.6	10.1	3.1	12.4	8.8
	MF	2.5	82.4	13.1	1.8	*	2.8	83.5	11.1	2.1	*
		1.5	**84.7**	11.5	2.2	*	2.5	**84.2**	10.8	2.1	*
		2.6	74.7	14.8	3.2	4.7	3.0	77.0	11.8	4.5	3.7
	GB	1.9	10.3	**83.8**	3.6	*	1.9	10.8	83.4	3.3	*
		*	11.4	81.9	6.1	*	1.3	8.3	**83.6**	6.5	*
		*	13.6	74.8	10.0	1.1	*	15.2	75.9	7.5	1.0
	RS	11.2	2.9	4.6	**80.3**	1.0	11.5	3.0	4.1	79.8	1.6
		12.4	2.6	9.7	75.2	*	12.7	1.6	4.5	**80.9**	*
		13.5	5.2	15.4	57.2	8.8	13.9	6.8	10.9	60.6	7.8
	JP	1.8	*	*	1.1	96.0	1.8	*	*	1.4	95.8
		*	*	*	*	**99.5**	*	*	*	*	**99.4**
		7.1	4.2	*	7.2	81.0	4.9	2.6	*	4.8	86.9
80	Unaltered	**66.3**	10.8	8.2	7.8	6.9	**66.3**	9.5	7.4	8.4	8.4
		62.2	7.9	7.2	18.2	4.5	63.9	7.1	5.6	19.3	4.1
		48.0	11.5	7.9	12.3	20.3	43.3	11.4	5.8	16.8	22.7
	MF	4.1	73.5	17.7	3.0	1.7	4.9	73.8	15.6	3.4	2.3
		2.6	**74.7**	16.5	4.8	1.4,	4.1	**77.1**	12.8	4.6	1.4
		2.1	62.6	20.2	7.2	7.9	2.4	64.9	16.4	7.3	9.0
	GB	4.0	19.2	**70.5**	4.0	2.3	4.5	17.4	**71.8**	3.4	2.9
		1.5	17.3	69.3	10.5	1.4	1.9	16.2	70.8	9.4	1.7
		*	22.1	61.8	11.7	3.8	*	18.4	67.5	9.0	4.4
	RS	9.8	5.1	5.6	**76.8**	2.7	11.0	4.5	4.8	**76.2**	3.6
		14.2	5.7	13.1	64.3	2.7	16.1	5.6	10.3	65.4	2.6
		10.4	11.8	18.2	44.7	14.9	9.6	10.6	16.0	47.0	16.8
	JP	12.6	3.5	2.9	3.4	77.6	12.4	3.1	2.9	3.3	78.3
		6.5	1.7	1.3	3.6	**86.9**	6.6	1.4	1.6	3.9	**86.5**
		20.1	12.6	4.3	9.9	53.1	16.2	9.9	3.0	11.4	59.5

Because the proposed feature set $F_{ARCPACTP}$ consists of F_{ARC}, F_{PAC} and F_{TP}. We then compare $F_{ARCPACTP}$ with its subsets to show the complementary effects of subsets. As shown in Tab. 2, F_{TP} and F_{ARC} obtain unsatisfactory results on "Unaltered VS RS" (84.5%) and "Unaltered VS JP" (73.4%), respectively. The combination F_{TP}+F_{ARC} improves the *Acc* on the above two classifications. The combination of newly proposed F_{PAC} on F_{TP}+F_{ARC} further improve the performance on "Unaltered VS RS" test, which indicates that F_{ARC}, F_{PAC} and F_{TP} complement each other. In order to verify that each feature subset is indispensable, we also test the effectiveness of the features by removing one subset from $F_{ARCPACTP}$, i.e., F_{TP} +F_{ARC}, F_{TP}+F_{PAC}, and F_{PAC}+F_{ARC}. The results in Tab. 2 demonstrate that $F_{ARCPACTP}$ performs better than F_{TP} +F_{AR}, F_{TP}+F_{PAC}, and F_{PAC}+F_{AR} in most of tests. These results again indicate that each feature subset plays its own role and the proposed $F_{ARCPACTP}$ gets the advantage from them.

Table 4: Average accuracy results (%) for identifying multi-class manipulations using different methods on **baseline test**. Best results are displayed in bold

Training, Tesing	Image size	JPEG QF	Dense-CNN	MISLnet	Proposed
Training: Bossraw	128×128	90	**95.5**	95.0	95.2
Testing: Bossraw		80	**90.4**	88.2	88.6
	32×32	90	84.3	71.4	**84.4**
		80	71.5	54.0	**72.9**
	16×16	90	72.6	--	**73.9**
		80	58.7	--	**61.8**
Training: Raiseraw	128×128	90	**96.3**	94.4	95.4
Testing: Rasieraw		80	**90.6**	88.2	89.1
	32×32	90	**85.8**	73.2	84.3
		80	72.7	56.4	**73.3**
	16×16	90	72.9	--	**73.8**
		80	58.2	--	**61.6**

3.3 Performance comparisons with prior arts

In this sub-section, we compare the proposed method with MISLnet [Bayar and Stamm (2018)] and Dense-CNN [Chen, Kang, Shi et al. (2019)] for multiple manipulations identifications under JPEG 90 and JPEG 80 compressions. For brevity, we only report the confusion matrices for image of size 32×32. It can be seen from Tab. 3 that the proposed method achieves the best identification rates for GB and RS on both Bossraw and Raiseraw database, while the Dense-CNN performs best for identifying JP and MF. In all tests, identifying JP is the easiest, while identifying RS and Unaltered image are the most difficult. As GB and MF are both smoothing filters, they are easily misclassified for each other. Through analyzing the results, we find that it can improve the performance of multi-classifier by improving the performances of difficult binary classifier including "Unaltered VS RS" and "GB VS MF", which is our future work.

To show the overall performance, the average result which is the average value of diagonal elements of confusion matrix is also given in Tabs. 4 and 5. Tab. 4 shows the results of baseline test, where testing images and training images are from the same image source. It is shown that Dense-CNN performs the best for images of size 128×128, while the proposed method performs best in most of tests for other two resolutions, and exhibits advantages in the robustness against JPEG compression. Taken the JPEG 80 compressed images of size 16×16 for example, the proposed method achieves 3.1% and 3.4% higher of average accuracy on Bossraw and Raiseraw database than Dense-CNN. Tab. 5 show the results of generalization ability test, where testing images and training images are from different image source. Specially, Bossraw and Raiseraw alternate as training set and test set. For comparative purposes, the testing set is set as the same with that in base line test. Tab. 5 shows that, due to mismatch between training source and testing source, almost all accuracy results decrease, but the proposed method has less declines. Specially, the decrease of *Acc* for Dense-CNN, MISLnet, and proposed method are within [1.3%, 3.4%], [-0.3%, 4.6%] and [0.4%, 1.6%] respectively. As the results of baseline test, the proposed method performs best for the JPEG compressed images of size 32×32 and 16×16, and the advantages against other two methods are increasing, which indicates that the proposed multi-classifier owns better generalization ability.

We also show the performance comparisons for mixed JPEG compression QFs on Bossraw database. In some cases, we probably have no prior knowledge about JPEG QF, nor can estimate it accurately. To solve this problem, a multi-classifier is trained from a variety of QFs, instead of a specific QF. Specially, the multi-classifier is trained on a mixed JPEG {90, 80}, which is composed by uniformly selected JPEG 90 and JPEG 80 compressed images. The proposed method/Dense-CNN obtains an average accuracy of 91.0%/91.0%, 76.8%/76.5%, and 66.0%/62.8 for 128×128, 64×64, and 32×32 image respectively. These results are a little lower than the average values of specific JPEG 90 and JPEG 80 compression detection results. Therefore, training the multi-classifier on JPEG compressed images with various QFs is probably useful when having no prior knowledge about JPEG settings.

Overall, the proposed F_{ARPACTP} performs better than the Dense-CNN and MISLnet in most of tests for JPEG compressed image of size 32×32 and 16×16. Besides, the proposed method is faster and less computing resources needed. For an image of size 32×32, the proposed method only spends about 0.018 seconds for extracting feature under settings: Intel Core i7 3.4 GHz CPU+16G RAM.

4 Conclusion

In this paper, we develop a multi-purpose forensic scheme for identifying multiple image manipulations. The main contributions are as follows:

(1) A universal feature set is proposed. The novelty lies that the partial correlation coefficient is proposed to purely measure neighborhood correlations. Combining with autoregressive coefficient and transition probability, the proposed feature can measure how manipulations change the neighborhood relationships in both linear and non-linear way. After a series of dimension reductions, the proposed feature set can accelerate the training and testing for the multi-purpose detector.

(2) The proposed scheme outperforms state-of-the-arts for JPEG compressed image of low resolution. Experimental results of the baseline test and generalization test on different databases demonstrate that the proposed method is effective and stable for the multi-purpose forensics. Further, the results for the JPEG compressed image of size 16×16 show that the proposed method achieves at least 3.1% improvement in term of average accuracy when compared with state-of-the-arts.

It is worthy to notice that the results of multi-purpose forensics for the JPEG compressed low resolution image are far from some practical applications, such as block-based tampering detection. Besides, a more efficient multi-classification scheme is needed when more image manipulations are considered.

Table 5: Average accuracies (%) for identifying multi-class manipulations using different methods on **generalization ability test**. Best results are displayed in bold

Training, Tesing	Image size	JPEG QF	Dense-CNN	MISLnet	Proposed
Training: Raiseraw	128×128	90	94.0	92.8	**94.3**
Testing: Bossraw		80	**87.4**	84.9	87.1
	32×32	90	82.8	71.3	**83.3**
		80	69.8	54.3	**71.5**
	16×16	90	71.3	--	**72.9**
		80	57.0	--	**60.9**
Training: Bossraw	128×128	90	**94.7**	93.3	94.3
Testing: Rasieraw		80	**89.1**	87.0	87.5
	32×32	90	**83.6**	69.6	83.2
		80	69.3	51.8	**71.7**
	16×16	90	71.0	--	**72.6**
		80	56.7	--	**60.5**

Funding Statement: This work was partially supported by NSFC (No. 61702429), Sichuan Science and Technology Program (No. 19yyjc1656).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Bayar, B.; Stamm, M. C (2018): Constrained convolutional neural networks: a new approach towards general purpose image manipulation detection. *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 3, pp. 1556-6013.

Bas, P.; Filler, T.; Pevný, T. (2011): Break our steganographic system-the ins and outs of organizing BOSS. *Information Hiding Conference*, vol. 6958, LNCS, pp. 59-70.

Chen, C.; Shi, Y.; Su, W. (2009): A machine learning based scheme for double JPEG compression detection. *International Conference on Pattern Recognition*, vol. 3, pp. 13-17.

Chen, Y.; Kang, X.; Shi, Y.; Wang, Z. (2019): A multi-purpose image forensic method using densely connected convolutional neural networks. *Journal of Real-Time Image Processing*, vol. 16, no. 3, pp. 725-740.

Cao, G.; Zhao, Y.; Ni, R.; Kot, A. (2011): Unsharp masking sharpening detection via overshoot artifacts analysis. *IEEE Signal Processing Letters*, vol. 18, no. 10, pp. 603-606.

Chen, C.; Ni. J.; Shen, Z.; Shi, Y. (2017): Blind forensics of successive geometric transformations in digital images using spectral method: theory and applications. *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2811-2824.

Chen, J.; Kang, X.; Liu, Y.; Wang, Z. (2015): Median filtering forensics based on convolutional neural networks. *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1849-1853.

Dang-Nguyen, D.; Pasquini, C.; Conotter, V.; Giulia, B. (2015): RAISE-A raw images dataset for digital image forensics. *ACM Multimedia Systems Conference ACM*, vol. 2, pp. 219-224.

Fan, W.; Wang, K.; Cayre, F. (2015): General-purpose image forensics using patch likelihood under image statistical models. *IEEE International Workshop on Information Forensics and Security*, vol. 3, pp. 1-6.

Jeong, B. G.; Moon, Y. H.; Eom, I. K. (2015): Blind identification of image manipulation type using mixed statistical moments. *Journal of Electronic Imaging*, vol. 24, no. 1, pp. 23-34.

Kang, X.; Stamm, M. C.; Peng, A.; Liu, K. J. R. (2013): Robust median filtering forensics using an autoregressive model. *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 9, pp. 1456-1468.

Kang, Y. H.; Liu, F. L.; Yang, C. F.; Xiang, L. Y.; Luo, X. et al. (2019): Color image steganalysis based on channel gradient correlation. *International Journal of Distributed Sensor Networks*, vol. 15, no. 5, 1550147719852031.

Luo, S.; Peng, A.; Zeng, H.; Kang, X.; Liu, L. (2019): Deep residual learning using data augmentation for median filtering forensics of digital images. *IEEE Access*, vol. 7, pp. 80614-80621.

Luo, W.; Huang, J.; Qiu. G. (2010): JPEG error analysis and its applications to digital image forensics. *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 3, pp. 480-491.

Li, H.; Luo, W.; Qiu, X.; Huang, J. (2018): Identification of various image operations using residual-based features. *IEEE Transactions on Circuits System and Video Technology*, vol. 28, no. 1, pp. 31-45

Liu, Y.; Guan, Q.; Zhao, X.; Cao, Y. (2018): Image forgery localization based on multi-scale convolutional neural networks. *ACM Workshop on Information Hiding and Multimedia Security*, pp. 85-90.

Mendialdua, I.; Echegaray, G.; Rodriguez, I.; Lazkano, E.; Sierra, B. (2016): Undirected cyclic graph based multiclass pair-wise classifier: Classifier number reduction maintaining accuracy. *Neurocomputing*, vol. 171, pp.1576-1590.

Mauro, B.; Ehsan, N.; Benedetta, T. (2018): Detection of adaptive histogram equalization robust against JPEG compression, *International Workshop on Biometrics and Forensics*, vol. 4, pp. 1-8.

Peng, A.; Kang, X. (2016): Median filtering forensics based on multi-directional difference of filtering residuals. *Jisuanji Xuebao/Chinese Journal of Computers*, vol. 39, no. 3, pp. 503-515.

Pevný, T.; Bas, P.; Fridrich, J. (2010): Steganalysis by subtractive pixel adjacency matrix. *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 215-224.

Peng, F.; Lin, Z. X.; Zhang, X.; Long, M. (2019): Reversible data hiding in encrypted 2D vector graphics based on reversible mapping model for real numbers. *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 9, pp. 2400-2411.

Stamm, M. C.; Liu, K. J. R. (2010): Forensic estimation and reconstruction of a contrast enhancement mapping. *IEEE International Conference on Acoustics, Speech and Signal*, pp. 1698-1701, Dallas, Texas, USA.

Steven, K. M. (1988): *Modern Spectral Estimation: Theory and Application*. Englewood Cliffs, NJ: Prentice Hall.

Wang, J.; Zhang, Y. (2020): Median filtering forensics scheme for color images based on quaternion magnitude-phase CNN. *Computers, Materials & Continua*, vol. 62, no. 1, pp. 99-112

Yang, J.; Ren, H.; Zhu, G.; Huang, J.; Shi, Y. (2018): Detecting median filtering via two-dimensional AR models of multiple filtered residuals. *Multimedia Tools and Applications*, vol. 77, no. 7, pp. 7931-7953

Zhang, Q.; Xiao, H.; Xue, F.; Lu, W.; Liu, H. et al. (2019): Digital image forensics of non-uniform deblurring. *Signal Processing: Image Communication*, vol. 76, pp. 167-177.