Discrete Cosine Transform Residual Feature based Filtering Forgery and Splicing Detection in JPEG Images

Aniket Roy, Diangarti Bhalang Tariang, Rajat Subhra Chakraborty
Department of Computer Science and Engineering,
Indian Institute of Technology Kharagpur
Kharagpur, West Bengal, India 721302
{ank.roy4,diazz.tariang.89,rschakraborty}@gmail.com

Ruchira Naskar
Department of Computer Science and Engineering
National Institute of Technology, Rourkela
India - 769008.
ruchira.naskar@gmail.com

Abstract

Digital images are one of the primary modern media for information interchange. However, digital images are vulnerable to interception and manipulation due to the wide availability of image editing software tools. Filtering forgery detection and splicing detection are two of the most important problems in digital image forensics. In particular, the primary challenge for the filtering forgery detection problem is that typically the techniques effective for non-linear filtering (e.g. median filtering*) detection are quite ineffective for linear filtering detection, and vice versa. In this paper, we have used* Discrete Cosine Transform Residual *features to train a* Support Vector Machine *classifier, and have demonstrated its effectiveness for both linear and non-linear filtering (specifically,* Median Filtering*) detection and filter classification, as well as re-compression based splicing detection in JPEG images. We have also theoretically justified the choice of the abovementioned feature set for both type of forgeries. Our technique outperforms the state-of-the-art forensic techniques for filtering detection, filter classification and re-compression based splicing detection, when applied on a set of standard benchmark images.*

1. Introduction

Digital multimedia forensics is closely related to several biometric applications in motivation and usage of computational tools, where the authenticity and fidelity of multimedia contents need to be verified [1]. In today's media saturated world, digital images are the most popular form of multimedia for information exchange. But due to easy availability of highly sophisticated and easy-to-use image processing software (often free), these multimedia contents are highly vulnerable to interception and manipulation. Hence, image forgery detection is one of the most active research areas in digital forensics.

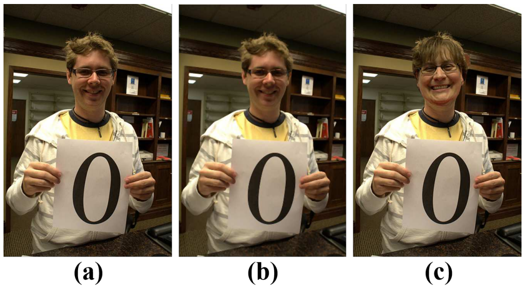

Figure 1. Example of biometric image tampering: (a) Original image; (b) Median filtered image; (c) Face spliced image.

In the literature on digital image forensics, there exist much work on *copy-move* and *copy-paste* forgery detection and localization approaches [2]. However, comparatively less work has been done for image filtering forgery detection. Moreover, these works also lack proper mathematical justification for the choice of features for image classification. Generally, forgeries are followed by an enhancement technique to make the forgery more convincing and less detectable. The enhancement techniques might include noise reduction, filtering, contrast enhancement, deblurring, edge sharpening, etc.; but among these techniques filtering is the most common and widely used. Hence, detection of whether an image had undergone such filtering enhancement is important for forensic analysis, and complements the content forgery detection techniques (e.g. those that detect copy-move forgery). Another very common form of image forgeries is splicing, where some parts of an image is forged by replacing them with parts from different image sources. JPEG images when undergoing such forgeries requires to be re–saved. The tampered image when re–saved as a JPEG file usually undergoes re–compression, resulting in the periodic effects on the re–compressed DCT coefficients [3]. Fig. 1 shows examples of median-filtered image and face splicing forgery, which are threats to the security of generic biometric authentication systems.

Recently, filtering forgery detection, especially median filtering detection has gained attention of the researchers [4–6]. It has been found that forensic techniques that are otherwise effective in detecting linear operations (such as sampling or scaling) on images [6], are often unable to detect non-linear filtering (such as median filtering) satisfactorily. In [4], Kirchner et al. used *Subtractive Pixel Adjacency Matrix* (SPAM) features in the first-order difference domain to detect median filtering for both uncompressed and JPEG images. Zhang et al. [5] utilized high-order local ternary pattern features for median filtering detection, exploiting the fact that median filtering produces regions of constant or nearly constant intensity values. Autoregressive model based median filtering detection has been proposed by Kang et al. [6]. Recently, more effective combined *Global Probability and Local Correlation Features (GLF)* [7], linear and non-linear descriptors [8] were proposed, especially to detect median filtering; however, these techniques are not quite effective for linear filtering detection.

On the other hand, linear filtering and compression detection forensic techniques [9, 10] perform poorly with respect to median filtering detection in particular, and non-linear filtering detection in general. Therefore, effective detection of linear as well as non-linear filtering by the same technique is relatively rare in the literature. Ravi et al. [11, 12] used modified *Transition Probability Matrix* (TPM) feature to detect the linear as well as non-linear filtering in both uncompressed and JPEG compressed images. However, the work lacked rigorous justification of effectiveness of the feature for both linear and non-linear filtering detection, as non-linear filtering is practically difficult to model analytically. Also, the effectiveness of the proposed features for detection of other types of attacks was not explored.

For image splicing detection, several techniques have been proposed so far. Shi et al. [13] proposed statistical features based on 1-D and 2-D moments, and transition probability features based on Markov chain in the DCT domain for image splicing detection. Huang et al. [14] used Markov features in DCT and DWT domains to identify splicing images. Fu et al. [15] exploited the *Hilbert-Huang Transform* (HHT) to generate the features needed for classification of forged and unforged images. The presence of the periodic effects on the re–compressed DCT coefficients can also be used to verify if the image has undergone tampering. Lin et al. in [3, 16] detected tampered images by examining the *Double Quantization* (DQ) effect on the DCT coefficients. In [17], the authors analyzed the experimental performance of the algorithm proposed in [3], on several variants of the CASIA TIDE (v.2.0) image dataset [18]. In [19], Markov random process was used to model the differences between the DCT coefficient and its connected elements. Fridrich et al. [20] proposed a method based on *Support Vector Machine* (SVM) classifiers with feature vectors formed by histograms of low-frequency DCT coefficients. In [21], distributions of DCT coefficents in the forged and unforged region were used as mixture model. By estimating the parameters of the mixture model, likelihood map for each DCT block of being doubly compressed was computed. But in general, these works lack proper justification for the choice of features.

In this paper, we have used *Discrete Cosine Transform Residual* (DCTR) features, originally proposed by Fridrich et al. [22] for steganalysis, for linear and non-linear (median) filtering detection and classification as well as re-compression based splicing detection in JPEG images. Robertson et al. [23] have shown that quantization noise in DCT compressed images is in general correlated. **The main insight behind the technique is that the correlation of the quantization noise is perturbed while performing any filtering and compression operation**. DCTR features exploit the first order statistics of the quantization noise residual obtained by the decompressed JPEG image, using 64 kernels of the Discrete Cosine Transform. Ravi et al. [11] had also used a similar concept to design modified TPM features, and the technique reported performs better than both the existing linear and non-linear (median) filtering detection techniques in the literature. However, in our work, we demonstrate that the proposed DCTR based filtering detection technique outperforms the scheme proposed in [11]. We have addressed both filtering detection as well as filter classification in the paper. Moreover, we have demonstrated that these features are also effective in splicing forgery detection, based on recompression of the forged image. We have demonstrated the effectiveness of the scheme for benchmark images from the *Dresden*, *CASIA TIDE* and *UCID* image databases. The rest of the paper is organized as follows. The proposed filtering detection framework and theoretical justification for it are described in Sec. 2. Experimental results to establish the efficacy of the proposed scheme are presented in Sec. 3. Conclusions are drawn Sec. 4, with directions for our future research efforts.

2. Forgery Detection: Methodology and Theoretical Justification

2.1. DCTR Feature Extraction

To generate the DCTR features, the original image in spatial domain is convolved with the 64 DCT basis patterns, each of size 8×8, to get 64 new undecimated DCT planes, i.e., DCT residuals. The final features are generated as histograms of the DCT residuals. The steps for DCTR feature extraction from an image are given below [22, 24]:

- The JPEG image is decompressed to spatial domain without quantizing the pixel values to $\{0, \cdots, 255\}$ to avoid any loss of information.
- The DCT basis patterns of size 8×8 are generated as $\mathbf{B}^{(k,l)} = \left(B_{mn}^{(k,l)}\right), 0 \leq m, n, k, l \leq 7$:

$$B_{mn}^{(k,l)} = \frac{w_k w_l}{4} \cos\frac{\pi k\,(2m+1)}{16} \cos\frac{\pi l\,(2n+1)}{16} \tag{1}$$

  where $w_0 = \frac{1}{\sqrt{2}}$, $w_i = 1$ for $i > 0$.
- The decompressed JPEG image $\mathbf{X}$ is convolved with each of the 64 DCT basis patterns $\mathbf{B}^{(k,l)}$, to generate a set of 64 undecimated DCT, each of which is denoted by $\mathbf{U}^{(k,l)}$ for the (k,l)-th DCT basis pattern as:

$$\mathbf{U}^{(k,l)} = \mathbf{X} * \mathbf{B}^{(k,l)}, \qquad 0 \leq k, l \leq 7 \tag{2}$$

- According to the 64 DCT modes (a, b), $0 \leq a, b \leq 7$, corresponding to each DCT basis pattern in each 8×8 DCT block, the filtered undecimated DCT image $\mathbf{U}^{(k,l)}$ is subsampled by a step-size of 8 to get 64 sub-images $\mathbf{U}_{a,b}^{(k,l)}$, as shown in Fig. 2.
- For each sub-image $\mathbf{U}_{(a,b)}^{(k,l)}$, the histogram feature is extracted as:

$$\mathbf{h}_{a,b}^{(k,l)}(x) = \frac{1}{|\mathbf{U}_{a,b}^{(k,l)}|} \sum_{u \in \mathbf{U}_{a,b}^{(k,l)}} \left[Q_T\left(|u|/q\right) = x\right], \tag{3}$$

  where Q_T is a quantizer with integer centroids $\{0, 1, \cdots, T\}$, q denotes the quantization step, and $[P]$ is the *Iverson Bracket*, which is equal to '0' when the statement P is false, and '1' when P is true. Here, q is dependent on the JPEG quality factor also [22].
- All the histogram features of the 64 sub-images $\mathbf{U}_{a,b}^{(k,l)}$ are merged and combined to obtain the histogram feature $\mathbf{h}^{(k,l)}$ of the filtered undecimated DCT image $\mathbf{U}^{(k,l)}$. This merging operation aids in dimensionality reduction because of statistical correlation between the histogram features of the sub-images, since images have similar statistical characteristics in symmetrical orientation. For example, since $\mathbf{U}^{(k,l)} = \mathbf{X} * \mathbf{B}^{(k,l)}$, and the sum of the elements of $\mathbf{B}^{(k,l)}$ is zero (these are DCT modes), for natural images $\mathbf{X}$, the distribution of $u \in \mathbf{U}_{(a,b)}^{(k,l)}$ will be approximately symmetrical, and centered at zero for all a, b.
- For each filtered image, sixty-four separate $(T+1)$-dimensional histogram feature sets could be obtained when the threshold for histogram is set to T. Then, these histogram features can be merged to form one histogram feature set with dimension $25 \times (T+1)$ using symmetry properties [22]. Finally, for all the 64 sub-images total feature dimension would be $64 \times 25 \times (T+1)$. Authors in [22] used DCTR feature dimension to be 8000 as a good compromise between performance and detectability for efficient steganalysis taking $T = 4$.

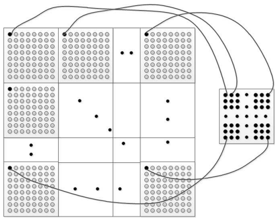

Figure 2. Subsampling procedure for DCTR feature extraction.

The primary advantage of the DCTR feature set is the relatively low computational complexity, and ability to provide better detection accuracy (for JPEG steganalysis) at relatively less feature dimension [22]. In this work, we have used the same feature set for filtering detection, filter classification and re-compression based splicing detection, as this feature involves first order statistics of quantized noise residuals.

2.2. Justification for Choice of Feature

In this section, we justify the effectiveness of DCTR features for both filtering and splicing forgery detection.

2.3. Filtering Forgery Detection using DCTR Features

We now theoretically justify the effectiveness of the DCTR features for filtering forgery detection. In our mathematical analysis, for ease of understanding and for a simpler notation, we would use an 1-D signal while presenting the operations. However, exactly similar mathematical arguments would hold for an 2-D signal like an image. Ravi

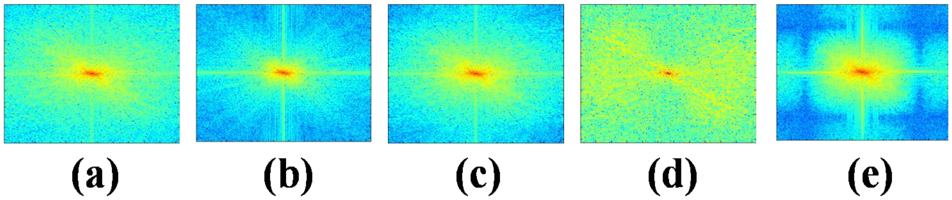

Figure 3. Power spectral density of the *Lena* image: (a) Original (Unfiltered); (b) Median filtered; (c) Gaussian filtered; (d) Laplacian filtered, and, (e) Average filtered.

et al. [12] tried to give a theoretical explanation for the binary classification problem of detecting whether an image is filtered or not, with an assumption that natural images follow the *Gaussian Mixture Model* (GMM) or the *Markov Random Field* (MRF) model. However, they do not justify theoretically the multi-class classification problem of detecting which type of filtering an image has undergone. In this paper, we have used DCTR features for both the binary classification and the multiple classification problem and observed that it outperforms the state-of-the-art TPM features. We will justify the efficacy of the DCTR feature for linear as well as non-linear (median) filtering.

The neighbouring pixel intensity differences in an image are captured by means of convolution with the 64 DCT basis functions through the generated DCTR features. Each class of image filtering operation imparts different impact on neighbouring intensity differences. For example, median filtering preserves edges and median filtered images exhibit regions of constant or nearly constant intensities [25]. Therefore, neighbouring intensity differences for median filtered images would likely to be very small values. Lowpass filtering is used for smoothing; hence, its corresponding neighbouring pixel intensity differences are also expected to be small quantities. On the other hand, highpass filters are used for sharpening and edge detection, which suggests that the corresponding neighbouring intensity differences for such filter should be relatively large values. Previously, Zhang et al. [5] used high-order local ternary patterns to find the traces for median filtering detection. Ravi et al. [11], [12] used transition probability based features to exploit the neighbouring intensity differences. However, we have observed via experimental results that DCT basis patterns more effectively exploits the neighbouring intensity differences through DCTR features.

2.3.1 Linear Filtering Model

Any linear filtering can be expressed as the convolution of the input image with a filter kernel [25]. Suppose, input image signal x is filtered with filter kernel l producing filtered image y. Then,

$$y = x * l \tag{4}$$

DCT basis functions contain several lowpass and highpass filters. A linear filter can be expressed as linear combination of these basis functions [25]. Now, suppose we extract the DCTR features from the filtered image y by convolving DCT basis function l', the resulting output is:

$$y' = y * l' = x * l * l' = x * A(l) \quad \text{if} \quad l' = l \tag{5}$$

where $A(l)$ represents the autocorrelation function of filter l. Now, if we take the Fourier Transform on both sides of Eq. (5), then it reduces to,

$$Y'(k) \leftrightarrow X(k) \cdot S(l) \tag{6}$$

where k denotes variable in transform domain, $S(l)$ denotes the power spectral density of filter l, $Y'(k)$ and $X(k)$ denotes the Fourier Transform of y' and x respectively. This utilizes the well known convolution multiplication property [25] of Fourier Transform and also that autocorrelation function and power spectral density are Fourier Transform pairs [25]. Since the DCT basis functions are symmetrical in nature, convolution with each basis pattern is similar to correlation with the 180° rotated version of that basis pattern. Thus, during DCTR feature extraction, the correlation profile of the filtered image with each of the DCT basis function is evaluated. The machine learning based model building selects exactly those components of the DCT basis function space which approximates the linear filter the image have been filtered with.

Fig. 3 shows the colormap plots of the power spectral densities of the original and filtered versions of the *Lena* image. As evident, the power spectral density of lowpass (Gaussian) and highpass (Laplacian) filtered images are distinctively different, and hence the DCTR feature characteristics obtained from the statistics of $Y'(k)$ are expected to be distinctive for different types of filters, as suggested by Eq.(6). However, this analysis is not readily applicable for non-linear filtering, as non-linear filtering cannot be expressed using linear convolution.

2.3.2 Non-linear Filtering Model

Non-linear filters are difficult to model as these filters do not often have explicit analytical forms. Among non-linear filters, median filters are most widely applied; hence we concentrate on modeling this type of non-linear filter.

Assuming $\{x_i\}$ be a stationary sequence with sample size n, marginal distribution function as $F(x)$ and density function to be $f(x)$. Now, median of the sequence, y_i can be approximately represented by the *Bahadur representation* [26] as:

$$\begin{aligned} y_i &= Median(x_{i-v}, \cdots, x_i, \cdots, x_{i+v}) \\ &\approx m + \tfrac{1}{2f(m)n} \sum_{p=-v}^{v} sign(x_{i+p} - m) \end{aligned} \tag{7}$$

where $v = (n-1)/2$ and m is chosen such that $F(m) = 0.5$. Generally, the expression holds for large sample size n; however it is equally applicable for small values of n also.

Now, an approximation formula for the covariance function of the median filtered sequence can be obtained by computing the covariance function for the moving average on the right hand side of Eq. (7):

$$r_y(\tau) \approx \frac{1}{nf^2(m)} \sum_{j=-(n-1)}^{n-1} \left(1 - \frac{|j|}{n}\right) c_{j+\tau} \tag{8}$$

where $c_k = \mathbf{Pr}(x_0 \leq m, x_k \leq m) - \frac{1}{4}$. For large n, Eq. (8) can be further approximated by,

$$r_y(\tau) \approx a \cdot \left(1 - \frac{|\tau|}{n}\right)^{+} \tag{9}$$

where a is a constant and $(b)^{+} = \max(b, 0)$. The power spectral density $S_y(k)$ is the Discrete Fourier Transform (DFT) of $r_y(\tau)$. Therefore,

$$\begin{aligned} S_y(k) &= \sum_{\tau=-n}^{n} r_y(\tau) \cdot \exp(-j2\pi k\tau) \\ &\approx \sum_{\tau=-n}^{n} a \cdot \left(1 - \frac{|\tau|}{n}\right)^{+} \cdot \exp(-j2\pi k\tau) \\ &= a\left[1 + 2 \cdot \sum_{\tau=1}^{n} \left(1 - \frac{\tau}{n}\right) \cdot \cos(2\pi k\tau)\right] \\ &= a\left[\tfrac{\sin^2(\pi kn)}{n \cdot \sin^2(\pi k)}\right] = a \cdot n \cdot \tfrac{\mathrm{sinc}^2(\pi kn)}{\mathrm{sinc}^2(\pi k)} \end{aligned} \tag{10}$$

where $\mathrm{sinc}(x) = \frac{\sin(x)}{x}$ is the *sinc function*, and the closed form result in the last step is obtained from [27]. A plot of ratio of these two $\mathrm{sinc}^2()$ functions itself resembles another periodic $\mathrm{sinc}^2()$ function (as can be easily verified using a plotting software). This can also be observed in the power spectral density of the median filtered image, consisting of repeated patterns with multiple peaks (resembling $\mathrm{sinc}^2()$ function) in its frequency response, as shown in the colormap in Fig. 3(b). Hence, we conclude that the DCTR features can distinguish median filtering also along with other linear filters, which is also evident from Fig. 3.

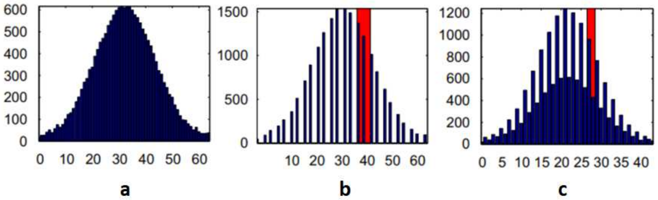

Figure 4. Example histogram of DCT coefficients in (a) single-compressed JPEG image; (b) and (c) double-compressed JPEG image. The case shown in (b) with periodically missing values happens when the first compression quality factor is not less than the second quality factor, and case (c) with periodic peaks and valleys happens when the first compression quality factor is less than the second quality factor. The shaded rectangles represent one period of the histograms.

2.4. Splicing Forgery Detection based Image Re-compression

In this section, we demonstrate that the DCTR features are also very effective in splicing forgery detection, based on the impact of image re-compression on the DCTR features. We consider the situation where the spliced image is re-saved in JPEG format, i.e., re-compression happens after splicing. Since, the DCTR features capture the quantization noise residuals, intuitively it is evident that such features should be capable of distinguishing re-compressed images from uncompressed images. We justify this observation both theoretically and experimentally. We will consider the following general steps for splicing forgery [3]: (a) decompression of the JPEG image; (b) replacement of a part of the image from another source image (may be JPEG compressed or not), and, (c) finally, saving (re-compressing) the tampered image in JPEG format. **Consequently, in the final forged JPEG image, the tampered regions have been quantized once, whereas the non-tampered regions have been quantized twice [3]. Therefore, the probability distribution function of the DCT coefficients of the non-tampered and tampered image should be distinguishable.** It has been established previously [3] that the double-quantized histogram of DCT coefficients, which are representative of their probability density function (pdf), have periodically missing values, or consists of periodic peaks and valleys, as shown in Fig. 4. Next we demonstrate that the pdf of the DCT coefficients of the uncompressed image is distinguishable with respect to the pdf of the re-compressed DCT coefficient.

As before, for simplicity we describe the analysis for an 1-D signal, but it is valid for any 2-D signal like an image. It is well-known that the distribution of the unquantized AC DCT coefficients can be modeled as a *Laplacian distribution* [23] of the form:

$$f(x) = \frac{\lambda}{2} \exp\left(-\lambda|x|\right) \tag{11}$$

where λ is a parameter. Quantization is a non-linear process. A statistical analysis of quantization process, presented in [28], reveals that the quantization process can be modeled as addition of independently and uniformly distributed random noise, with distribution parameter dependent on the quantization interval, to the quantized random variable. Addition of independent random variables implies convolution of their individual pdfs to get the pdf of the sum [29]. Therefore, the probability distribution of the quantized output (l) is the convolution of the input (x) pdf and the pdf the uniformly distributed noise (n):

$$f_l = f_x * f_n \tag{12}$$

Similarly, double quantization can also be modeled as addition of two uniformly (but with different distribution parameters) distributed independent random variables denoting the noise, with the input random variable, (as shown in Fig. 5) resulting in the pdf:

$$f_{l'} = f_x * f_n * f_{n'} \tag{13}$$

The pdf of the uniformly distributed noise for the single-quantized case (with quantization interval q_1) and the double-quantized case (with quantization interval q_1, followed by quantization interval q_2), are denoted as f_n and $f_{n'}$ respectively, and expressed as [28]:

$$f_n(x) = \begin{cases} \frac{1}{q_1}, & \text{if} -\frac{q_1}{2} < x \leq \frac{q_1}{2} \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

and,

$$f_{n'}(x) = \begin{cases} \frac{1}{q_2}, & \text{if} -\frac{q_2}{2} < x \leq \frac{q_2}{2} \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

According to the image tampering model [3], in the tampered image, the tampered regions are quantized once and the untampered regions are quantized twice. Therefore, the pdf of the DCT coefficient of the tampered image (f_t) could be modeled as:

$$f_t = w_1 f_l + w_2 f_{l'} = w_1 \cdot f_x * f_n + w_2 \cdot f_x * f_n * f_{n'} \tag{16}$$

where w_1 and w_2 are weights that represent the relative strengths of the single-quantized (tampered region) and double-quantized (non-tampered region) signal components respectively, in the tampered image. **Thus, the pdf of the tampered image DCT coefficients follow a Laplacian mixture model as shown in Eq. (16), in contrast to those of the untampered image which follow an ordinary Laplacian distribution.** Therefore, it is evident that the DCT coefficient distributions are distinguishable for untampered and tampered image, which can also be seen from Fig.4. Note that the quantization interval of the DCT coefficents are dependent on the quality factor of JPEG images [30]; hence, the analysis presented above can be easily adapted in terms of the JPEG image quality factor instead of the quantization intervals.

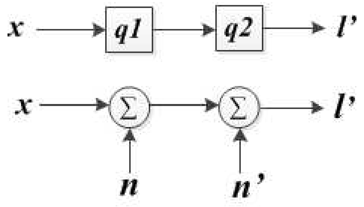

Figure 5. Double quantization model.

Table 1. Filters Used For Experiment

Filter Type (NL/L)	Kernel Size	Parameter Value
Median Filter (NL)	3×3	-
Gaussian Filter (L)	3×3	$\sigma = 0.5$
Average Filter (L)	3×3	-
Laplacian Filter (L)	3×3	$\alpha = 0.1$

Table 2. Filtering Detection Accuracy Comparison

Scheme	Filter Detection Accuracy			
	Median Filter	Gaussian Filter	Laplacian Filter	Average Filter
GLF feature based approach [7]	95.9%	63.1%	100%	99.0%
TPM feature based approach [11]	99.7%	82.5%	100%	99.9%
Proposed DCTR feature approach	100%	98.5%	100%	100%

In the next section, we validate the theoretical arguments presented above with experimental results.

3. Experimental Results

3.1. Experimental Setup

We have experimented with 500 pristine JPEG image taken by camera model **Sony H50** from the ***Dresden Image Database*** [31]. Subsequently, several linear and non-linear filtering operations as listed (along with their relevant parameters) in Table 1 were applied to that set of images to generate the filtered images. Our forensic experiments deal with both the filtering detection as well as filter classification problem. For re-compression based splicing detection, we evaluated on the *CASIA* tampered image detection evaluation database (**CASIA TIDE v2.0**) [18]. To illustrate the effectiveness of the proposed method over JPEG forgeries involving diverse compression ratios, we manually conducted the splicing attack on **UCID Image Database** [32]. All image processing and feature extraction operations were implemented using *Matlab* (v. 2015a). For machine learning based model building and validation, we used the *Weka Machine Learning Toolbox* [33].

Table 3. Filter Classification Accuracy Comparison

Scheme	Filtering Classification Accuracy
GLF feature approach [7]	80.36%
TPM feature approach [11]	89.4%
Our DCTR feature approach	99.32%

Table 4. Confusion Matrix For Filter Classification

Classific. Accuracy		**True Filter**				
		Unfiltered	**Median Filter**	**Gaussian Filter**	**Laplacian Filter**	**Average Filter**
Identified Filter	**Unfiltered**	**99%**	0%	1%	0%	0%
	Median Filter	0%	**100%**	0%	0%	0%
	Gaussian Filter	2.4%	0%	**97.6%**	0%	0%
	Laplacian Filter	0%	0%	0%	**100%**	0%
	Average Filter	0%	0%	0%	0%	**100%**

3.2. Experiment 1: Filtering Detection

The first experiment was carried out to detect whether a given image is filtered or not. For each of the filter mentioned in Table 1, 500 pristine JPEG images from the Dresden Image Database were filtered with that particular filter to generate a dataset of 1000 images, comprising of 500 non-filtered and 500 filtered images, to be used for binary classification of filtering detection. We use polynomial kernel based binary classification *Support Vector Machine* (SVM) classifier, and grid search was performed to determine the parameters that give better average accuracy using 10-fold cross-validation.

The performance of the state-of-the-art GLF feature based median filtering detection technique [7] degrades significantly for linear filtering detection, as shown in Table 2. However, our proposed technique outperforms [7] for both linear and median filtering detection, as evident from Table 2. Comparison for filtering detection with the state-of-the-art TPM feature based approach in [11] using single iteration is shown in Table 2, which reveals that our DCTR feature based filtering detection approach also outperforms the approach in [11] for each of the filters considered.

3.3. Experiment 2: Filter Classification

The more challenging task is to identify the type of filter applied. For that purpose, the same set of 500 pristine JPEG images from the Dresden Image Database was filtered with Median, Gaussian, Laplacian and Average filters as mentioned in Table 1, to generate five classes of images, namely: Unfiltered, Median-filtered, Gaussian-filtered, Laplacian-filtered and Average-filtered. Hence, the dataset for filter classification contains 2500 images with 500 images for each of the five classes. Then, the DCTR features are extracted from the images and used to train a multi-class SVM classifier. To avoid overfitting we have used 10-fold cross-validation in *Weka*.

Table 5. Splicing Detection Accuracy Comparison on CASIA Dataset

Scheme	Splicing Detection Accuracy
Markov DCT and DWT feaures [14]	89.76%
Lin et el. [17]	91.34%
Our DCTR feature approach	98.06%

Table 6. Splicing Detection Accuracy on Manually Tampered UCID Database

QF_2 QF_1		50	60	70	80	90
50	Proposed	86.03%	99.25%	100%	100%	100%
	[21]	93.4%	92.1%	88.2%	87.6%	90.5%
	[19]	-	100%	100%	100%	100%
60	Proposed	99.85%	96.32%	100%	100%	100%
	[21]	82.5%	84.4%	84.6%	84.8%	84.4%
	[19]	99%	-	100%	100%	100%
70	Proposed	100%	100%	98.06%	100%	99.9%
	[21]	86.7%	86.3%	85.6%	86.6%	86.2%
	[19]	100%	97%	-	100%	100%
80	Proposed	100%	100%	99.83%	97.48%	100%
	[21]	89.3%	88.6%	88.6%	89.3%	88.7%
	[19]	100%	99%	100%	-	100%
90	Proposed	99.6%	99.35%	100%	100%	99.07%
	[21]	68.4%	77%	72.7%	73.6%	70.8%
	[19]	100%	100%	99%	100%	-

Comparison for filter classification with the state-of-the-art GLF [7] and TPM feature based approach [11] using single iteration is shown in Table 3, which ensures that our DCTR feature based approach comfortably outperforms the approach in [7] and [11] for filter classification. The corresponding confusion matrix for filter classification for our approach is shown in Table 4. Hence, we infer the superiority of our approach to state-of-the-art individual linear [9] filterimg detection technique (since [11] ouperforms [9], and our proposed technique outperforms [11]), and median filtering [7] detection technique. It also outperforms the relatively recent generalized linear and median filtering detection and classification technique [11].

3.4. Experiment 3: Splicing Detection

The CASIA image dataset consists of 7491 authentic and 5123 tampered color images in JPEG, BMP, or TIFF formats. The images in this database are of different sizes, varying from 240 × 160 to 900 × 600 pixels. All experiments were conducted under the same experimental conditions as described in Section 3.2, where polynomial kernel based binary classification SVM classifier was used. Experimental results obtained on the *CASIA Tampered Image Detection Evaluation* (CASIA-TIDE) database (v 2.0) are shown in Table 5. For comparison, we tested the performance of the state-of-the-art methods [14, 17] on this dataset too. As evident from Table 5, our DCTR feature based splicing detection method outperforms the state–of–the–art methods with high detection accuracy results.

We further evaluate the performance of the proposed technigue on the UCID image database [32], containing

1338 uncompressed colour images (TIFF format) with size 512×384 or 384×512 pixels. JPEG compression of the UCID images at compression ratio QF_1 was performed using the `imwrite()` library function of MATLAB. Then, selected regions of the JPEG test images were forged using MATLAB scripts for our experiments, in the following way. An $m \times n$ region of the image was extracted, where $1 \leq m, n < 512$ and re-saved in JPEG format. The extracted region was later transplanted back to the same location of the original image, and the resultant tampered image was re-saved with a second quality factor QF_2. The size of the tampered region was 10%, 30% or 50% of the image size. The tampered images were then analyzed for forgery detection, and experimental results are tabulated in Table 6. We have compared our detection accuracy results with those presented in [19, 21]. The experimental results achieved for different ($QF1$, $QF2$) combinations are shown in Table 6. The proposed scheme outperforms the state-of-the-art technique [21] for all cases $QF_1 < QF_2$, $QF_1 = QF_2$ and $QF_1 > QF_2$, with detection accuracy close to 100%, as evident from Table 6. As compared to [19], our proposed technique achieves equally high detection accuracy. However, for the case $QF_1 = QF_2$, our technique outperforms that of [19], as is evident from Table 6.

4. Conclusion

We have proposed a DCTR feature based approach for both filtering forgery detection and classification as well as re-compression based splicing detection with SVM classifier. Effectiveness of DCTR features were theoretically justified for linear and non-linear (median) filtering detection and classification problems, along with re-compression based splicing detection. Experimentally, the efficacy of the proposed technique and its superiority over state-of-the-art techniques was also verified for several types of linear filtering, median filtering and splicing detection for different standard benchmark dataset. Our future research efforts would be directed towards exploring the applicability of the proposed technique for other image formats and also an unified forgery detection technique, effective for multiple image forgeries.

References

[1] A. K. Jain and A. Ross, “Bridging the gap: from biometrics to forensics,” *Phil. Trans. R. Soc. B*, vol. 370, no. 1674, p. 20140254, 2015. 1

[2] M. K. Johnson and H. Farid, “Exposing digital forgeries through chromatic aberration,” in *Proceedings of the 8th workshop on Multimedia and security*. ACM, 2006, pp. 48–55. 1

[3] Z. Lin, J. He, X. Tang, and C.-K. Tang, “Fast, automatic and fine-grained tampered jpeg image detection via dct coefficient analysis,” *Pattern Recognition*, vol. 42, no. 11, pp. 2492–2501, 2009. 2, 5, 6

[4] M. Kirchner and J. Fridrich, “On detection of median filtering in digital images,” in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2010, pp. 754 110–754 110. 2

[5] Y. Zhang, S. Li, S. Wang, and Y. Q. Shi, “Revealing the traces of median filtering using high-order local ternary patterns,” *IEEE Signal Processing Letters*, vol. 21, no. 3, pp. 275–279, 2014. 2, 4

[6] X. Kang, M. C. Stamm, A. Peng, and K. R. Liu, “Robust median filtering forensics using an autoregressive model,” *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 9, pp. 1456–1468, 2013. 2

[7] C. Chen, J. Ni, and J. Huang, “Blind detection of median filtering in digital images: A difference domain based approach,” *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 4699–4710, 2013. 2, 6, 7

[8] Z. Shen, J. Ni, and C. Chen, “Blind detection of median filtering using linear and nonlinear descriptors,” *Multimedia Tools and Applications*, vol. 75, no. 4, pp. 2327–2346, 2016. 2

[9] V. Conotter, P. Comesaña, and F. Pérez-González, “Forensic analysis of full-frame linearly filtered jpeg images,” in *2013 IEEE International Conference on Image Processing*. IEEE, 2013, pp. 4517–4521. 2, 7

[10] ——, “Joint detection of full-frame linear filtering and jpeg compression in digital images,” in *2013 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2013, pp. 156–161. 2

[11] H. Ravi, A. Subramanyam, and S. Emmanuel, “Spatial domain quantization noise based image filtering detection,” in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1180–1184. 2, 4, 6, 7

[12] ——, “Forensic analysis of linear and nonlinear image filtering using quantization noise,” *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 12, no. 3, p. 39, 2016. 2, 3, 4

[13] Y. Q. Shi, C. Chen, and W. Chen, “A natural image model approach to splicing detection,” in *Proceedings of the 9th workshop on Multimedia & security*. ACM, 2007, pp. 51–62. 2

[14] Z. He, W. Lu, W. Sun, and J. Huang, “Digital image splicing detection based on markov features in dct and dwt domain,” *Pattern Recognition*, vol. 45, no. 12, pp. 4292–4299, 2012. 2, 7

[15] D. Fu, Y. Q. Shi, and W. Su, “Detection of image splicing based on hilbert-huang transform and moments of characteristic functions with wavelet decomposition,” in *International Workshop on Digital Watermarking*. Springer, 2006, pp. 177–187. 2

[16] J. He, Z. Lin, L. Wang, and X. Tang, “Detecting doctored jpeg images via dct coefficient analysis,” in *European conference on computer vision*. Springer, 2006, pp. 423–435. 2

[17] G. Cattaneo, G. Roscigno, and U. F. Petrillo, “Experimental evaluation of an algorithm for the detection of tampered jpeg images,” in *Information and Communication Technology-EurAsia Conference*. Springer, 2014, pp. 643–652. 2, 7

[18] “Institute of Automation, Chinese Academy of Sciences (CASIA). CASIA Tampered Image Detection Evaluation Database (CASIA TIDE) v2.0,” 2013, available at http://forensics.idealtest.org/. 2, 6

[19] C. Chen, Y. Q. Shi, and W. Su, “A machine learning based scheme for double jpeg compression detection,” in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008, pp. 1–4. 2, 7, 8

[20] J. Lukáš and J. Fridrich, “Estimation of primary quantization matrix in double compressed jpeg images,” in *Proc. Digital Forensic Research Workshop*, 2003, pp. 5–8. 2

[21] W. Wang, J. Dong, and T. Tan, “Exploring dct coefficient quantization effects for local tampering detection,” *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 10, pp. 1653–1666, 2014. 2, 7, 8

[22] V. Holub and J. Fridrich, “Low-complexity features for jpeg steganalysis using undecimated dct,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 219–228, 2015, DCTR extraction code available online at http://dde.binghamton.edu/download/feature_extractors/. 2, 3

[23] M. A. Robertson and R. L. Stevenson, “Dct quantization noise in compressed images,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 1, pp. 27–38, 2005. 2, 5

[24] X. Song, F. Liu, C. Yang, X. Luo, and Y. Zhang, "Steganalysis of adaptive jpeg steganography using 2d gabor filters," in *Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security*. ACM, 2015, pp. 15–23. 3

[25] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd ed.)*. Prentice Hall, 2008. 4

[26] B. Justusson, "Median filtering: Statistical properties," in *Two-Dimensional Digital Signal Processing II*. Springer, 1981, pp. 161–196. 5

[27] C. Blatter, "DTFT of a triangle function in closed form," 2016, Mathematics Stack Exchange. Accessed: May 2017. [Online]. Available: https://math.stackexchange.com/q/119337 5

[28] B. Widrow, I. Kollar, and M.-C. Liu, "Statistical theory of quantization," *IEEE Transactions on instrumentation and measurement*, vol. 45, no. 2, pp. 353–361, 1996. 5, 6

[29] B. P. Lathi *et al.*, *Linear systems and signals*. Oxford University Press New York:, 2005, vol. 2. 6

[30] I. DIS, "10918-1. Digital Compression and Coding of Continuous-tone Still Images (JPEG)," *CCITT Recommendation T*, vol. 81, 1991. 6

[31] T. Gloe and R. Böhme, "The dresden image database for benchmarking digital image forensics," *Journal of Digital Forensic Practice*, vol. 3, no. 2-4, pp. 150–159, 2010. 6

[32] G. Schaefer and M. Stich, "UCID: an uncompressed color image database," in *Electronic Imaging 2004*. International Society for Optics and Photonics, 2003, pp. 472–480. 6, 7

[33] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009. 6