Forensic detection of Median filtering in Images using Local Tetra Patterns and J-Divergence

Udayeni Anumala, Manish Okade
Department of Electronics and Communication Engineering
National Institute of Technology
Rourkela, Odisha
email:udayeni.anumala2@gmail.com, okadem@nitrkl.ac.in

***Abstract*—This paper presents a novel application of local tetra patterns to the median filtering detection problem. The premise of the proposed method is based on the ability of the local tetra patterns in identifying the streaking fingerprints left over by the application of a median filter on an image. These streaking fingerprints serve as a clue in determining the authenticity of an image towards the application of a median filter. The streaking pixels are identified by establishing the relationship of every pixel with respect to its neighboring pixels. The relationship is in the form of horizontal and vertical derivative directions and magnitudes followed by the tetra pattern and magnitude assignment. The feature vector generated utilizing the local tetra patterns is reduced by using the J-divergence in-order to keep the computational complexity low. Experimental testing for the proposed method along with comparative analysis carried out with existing state-of-the-art methods shows good performance at reduced computational complexity for the proposed method.**

I. Introduction

Median filtering forensic detection is an important problem due to its notable characteristic of smoothing the image without disturbing it's edges thereby making it a popular choice during image tampering so that the forged area looks realistic. Due to its non-linear nature it may also be used to remove traces left over by other editing operations thereby making median filtering a popular choice for hiding forgeries.

Median filtering detection faces challenges like presence of JPEG post compression, presence of other filtering manipulations and small local areas being median filtered. Research effort towards mitigating these challenges for median filtering detection have previously focused on utilizing statistical properties of first-order difference of pixel values [1], analyzing probability of first order difference of pixels in textured regions [2], using local difference descriptor, utilizing subtractive pixel adjacency matrix (SPAM) as a feature vector [3], SPAM features coupled with ratio of histogram bins [4]. Features extracted from an images median filter residual (MFR) were used to detect median filtering by constructing a 1-D auto regressive (AR) model of the extracted MFR followed by utilizing a SVM trained on the AR features [5]. More recently deep learning based methods [6]–[10] have been investigated for this research problem with very high accuracy in comparison to traditional hand crafted feature based methods. However, although these methods achieve high accuracy yet they suffer from drawbacks namely adversarial attacks where the accuracy fails miserably if the training input is corrupted as well as the training for these networks needs dedicated hardware like GPUs. Inspite of utilizing GPUs they need training time in hours which makes it highly time consuming. Median filtering has also found its applications in anti-forensics since it has the ability of hiding traces of image resampling as well as JPEG blocking artifacts [11]. Zhang et al. [12] analyzed the median filtering detection problem by applying texture analysis utilizing the local ternary pattern (LTP) coupled with the local derivative pattern (LDP) to form the second-order local ternary pattern which was utilized to detect the traces of median filtering. Since the feature vector employed was of very high dimension, kernel principal component analysis (KPCA) was used for reducing the dimensionality. Although this method outperformed existing methods yet had few shortcomings which form the motivation of the current work. Firstly, the LTP had three valued coding function which was unable to capture the derivative directions thereby leading to false positives. Secondly, it was computationally expensive due to the utilization of KPCA. Motivated by these shortcomings, this paper investigates the application of a powerful texture descriptor, the local tetra pattern (LTrP) [13] to the median filtering detection problem which as shown by our study mitigates the problems present in the 2nd order local ternary pattern as proposed by Zhang et al. [12]. To keep the computational complexity low we propose utilizing the J-Divergence which reduces the feature dimension at the same time maintains superior detection performance. Experimental validation carried on images from the UCID v2 dataset [14] along with comparative analysis with Zhang et al. method [12] shows superior detection accuracy coupled with reduced computational cost for the proposed method. We also compare with one of the popular deep learning based methods [9] to show the effectiveness of the proposed method.

II. Review of Local Tetra Pattern

The fundamental concept of LTrP [13] is to capture the image structure spatially by encoding the relationship of the center pixel (g_c) with respect to its neighboring pixels centered around a window of size $w \times w$. Assuming a 3×3 window the center pixel g_c will have 8 neighbors as shown below

g_1	g_2	g_3
g_8	g_c	g_4
g_7	g_6	g_5

For an image denoted by I, the first order derivatives (I^1) along 0^o and 90^o directions are calculated using $I^1_{0^o}(g_c) = I(g_4) - I(g_c)$ and $I^1_{90^o}(g_c) = I(g_2) - I(g_c)$ respectively.

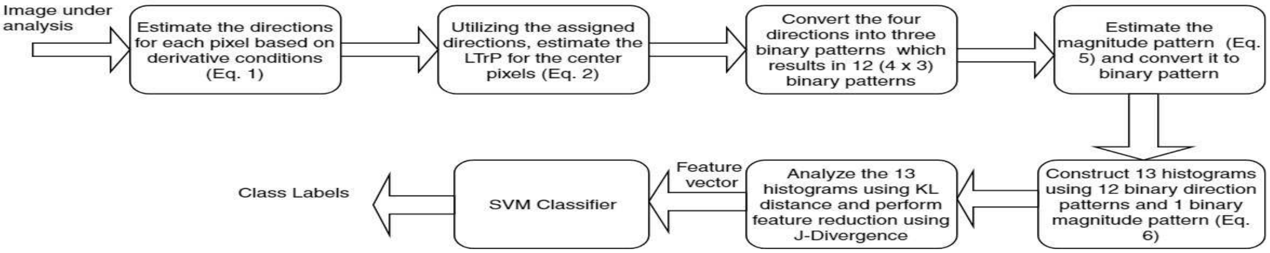
Image under analysis
Estimate the directions for each pixel based on derivative conditions (Eq. 1)
Utilizing the assigned directions, estimate the LTrP for the center pixels (Eq. 2)
Convert the four directions into three binary patterns which results in 12 (4 x 3) binary patterns
Estimate the magnitude pattern (Eq. 5) and convert it to binary pattern
Construct 13 histograms using 12 binary direction patterns and 1 binary magnitude pattern (Eq. 6)
Analyze the 13 histograms using KL distance and perform feature reduction using J-Divergence
Feature vector
SVM Classifier
Class Labels

Fig. 1: Overview of proposed median filtering detection system

Utilizing this the direction of center pixel is calculated using

$$I^1_{\text{Dir.}}(g_c) = \begin{cases} 1, & \text{if } I^1_{0^o}(g_c) \geq 0 \text{ and } I^1_{90^o}(g_c) > 0 \\ 2, & \text{if } I^1_{0^o}(g_c) < 0 \text{ and } I^1_{90^o}(g_c) \geq 0 \\ 3, & \text{if } I^1_{0^o}(g_c) < 0 \text{ and } I^1_{90^o}(g_c) < 0 \\ 4, & \text{if } I^1_{0^o}(g_c) \geq 0 \text{ and } I^1_{90^o}(g_c) < 0 \end{cases} \tag{1}$$

Eq. (1) implies that each center pixel can take one among the four values i.e. 1, 2, 3 or 4 thereby converting the image window into four possible directions. Utilizing this computed direction information, the LTrP of order 2 for center pixel is calculated as

$$LTrP^2(g_c) = \Big\{ f_1(I^1_{\text{Dir.}}(g_c), I^1_{\text{Dir.}}(g_1)), f_1(I^1_{\text{Dir.}}(g_c), I^1_{\text{Dir.}}(g_2)), \ldots, f_1(I^1_{\text{Dir.}}(g_c), I^1_{\text{Dir.}}(g_P)) \Big\}\Big|_{P=8} \tag{2}$$

$$f_1(I^1_{\text{Dir.}}(g_c), I^1_{\text{Dir.}}(g_p)) = \begin{cases} 0, & \text{if } I^1_{\text{Dir.}}(g_c) = I^1_{\text{Dir.}}(g_p) \\ I^1_{\text{Dir.}}(g_p), & \text{elsewhere} \end{cases}$$

Utilizing Eq. (2), the 8-bit tetra pattern for each center pixel is obtained. The obtained tetra patterns (8-bit) are converted into 3 binary patterns based on the direction of center pixel, as follows

$$LTrP^2|_{Dir_{i \neq j}} = \sum_{p=1}^{P} 2^{(p-1)} \times f_2(LTrP^2(g_c))\big|_{Dir_{i \neq j}} \tag{3}$$

where, Dir.$= [1, 2, 3, 4]$

$$f_2(LTrP^2(g_c))|_{\phi} = \begin{cases} 1, & \text{if } LTrP^2(g_c) = \phi \\ 0, & \text{elsewhere} \end{cases}$$

where, ϕ indicates $i \neq j$ in Eq. (3). In addition to the 12 binary patterns a 13^{th} binary pattern known as magnitude pattern (MP) was also explored by utilizing the magnitudes of horizontal and vertical first order derivatives i.e.

$$M_{I^1(g_p)} = \sqrt{(I^1_{0^o}(g_p))^2 + (I^1_{90^o}(g_p))^2} \tag{4}$$

$$MP = \sum_{p=1}^{P} 2^{(p-1)} \times f_3(M_{I^1(g_p)} - M_{I^1(g_c)}) \tag{5}$$

$$f_3(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{elsewhere} \end{cases}$$

III. Proposed Method

Fig. 1 shows the overview of proposed median filtering detection system. The image under investigation is firstly analyzed in a localized manner to detect the presence of streaking fingerprints [15] by establishing relationship for every pixel with respect to its neighboring pixels. This relationship is in the form of directions and magnitudes of the first order derivatives along 0^o and 90^o. The direction assignment for each pixel is carried out based on the 0^o and 90^o derivative conditions as given by Eq. (1). Utilizing the assigned directions, the tetra pattern is calculated for the center pixel using Eq. (2). The four tetra patterns obtained are separated into three binary patterns each thereby resulting in 12 (4×3) binary patterns. The next step is to estimate the magnitude information for each pixel which is carried out based on the magnitude of the derivatives along 0^o and 90^o as given by Eq. (4). The magnitude pattern is next converted into binary pattern. Hence a total of 13 binary patters (12 direction patterns and 1 magnitude pattern) are formed which are converted into their histogram representations to obtain 13 histograms. These 13 histograms are compared using the KL divergence [16] to form the distance matrix. The J-Divergence [17] is then utilized to reduce the dimensionality of the matrix and the feature vector is formed. The feature vector is then fed to SVM classifier which detects the presence/absence of streaking fingerprints thereby indicating the traces of median filtering on the image under investigation. The detailed description of each stage is given below.

A. *Image analysis using LTrP*

Fig. 2 shows the analysis carried out on an unaltered 8×8 region referred as original region as well as after application of a median filter of window size 5×5 on the original region to obtain what is referred as the median filtered region. The first step is to obtain first order derivatives along 0^o and 90^o and are shown in Fig. 2(b) and Fig. 2(c). This is followed by assigning the center pixel with one of the four possible directions i.e. 1, 2, 3 or 4 by utilizing the first order derivative conditions i.e. Eq. (1) which is shown in Fig. 2(d). The magnitude of the first order derivatives along 0^o and 90^o is also estimated using Eq. (4) and is shown in Fig. 2(e). Utilizing the direction and magnitude components of the derivatives gives better detection performance as validated in the experimental results section. To demonstrate the streaking fingerprints left over by the application of median filtering, a 5×5 median filter is applied on the original 8×8 region. As observed, streaks (i.e. runs with pixel values 73, 74, 77 etc.) appear in the median filtered region

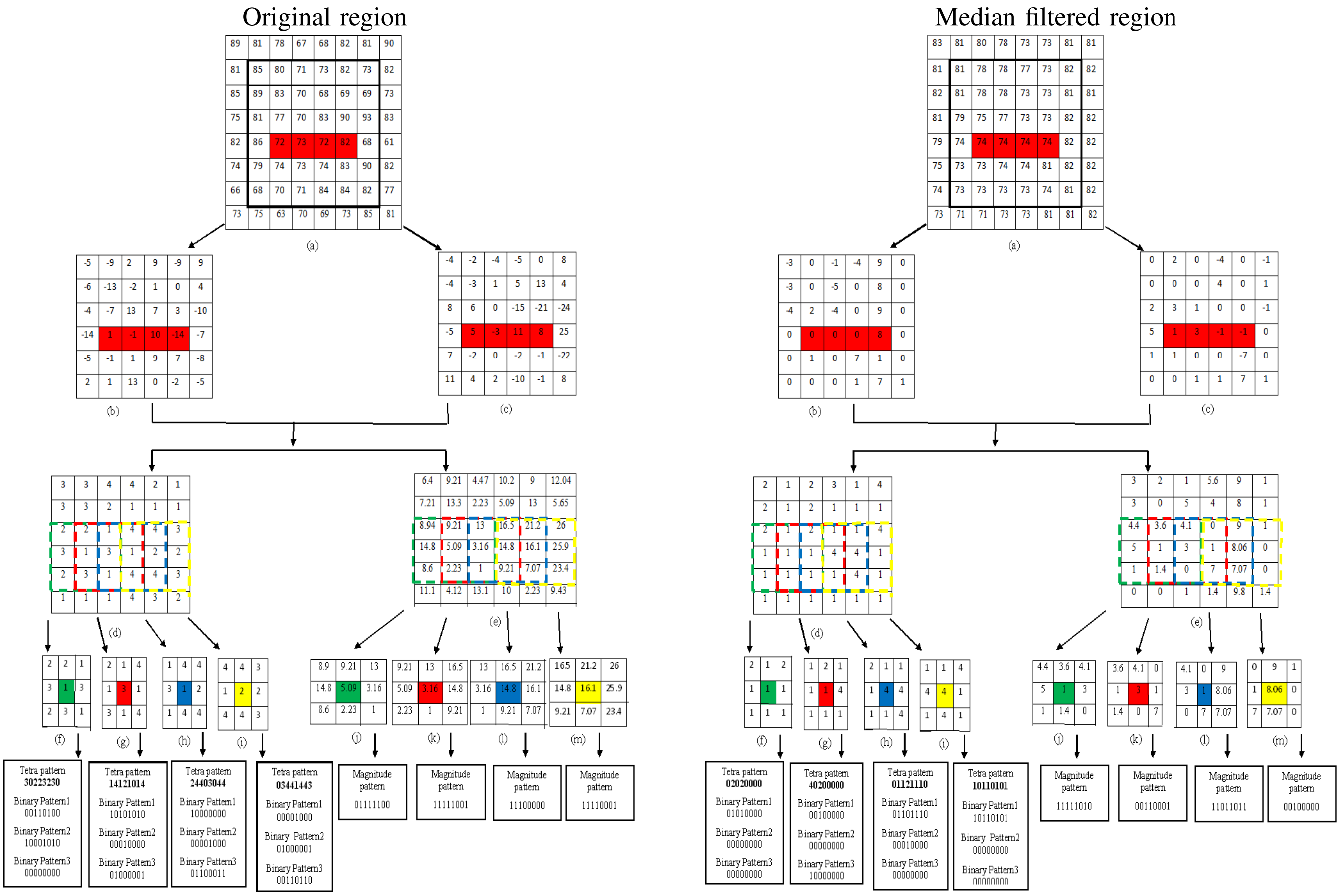

Fig. 2: (b) First order derivative along 0^o. (c) First order derivative along 90^o. (d) Direction assigned to pixels (Eq. (1)). (e) Gradient magnitude (Eq. (4)). (f)-(g)-(h)-(i) shows 3×3 neighborhood of streak pixels and their corresponding tetra patterns. (j)-(k)-(l)-(m) shows gradient magnitude of 3×3 neighborhood of streak pixels and their corresponding magnitude pattern.

which are the fingerprints left behind by the median filtering operation. We demonstrate the streaking detection using the streak with four number of runs with pixel intensity 74 (shown shaded). The 3×3 region around the streak is analyzed and is shown from Fig. 2(f)-(m) for the original region and the median filtered region. LTrP estimation (i.e. assignment of Tetra pattern and Magnitude pattern) is carried out and as observed the streaking pixels have different tetra as well as magnitude patterns in comparison to its counterpart in the original image which serves as the detection fingerprints. The four directions are converted into three binary patterns which leads to 12 (4×3) binary patterns. The magnitude pattern is also converted into its binary pattern thereby leading to 13 binary patterns.

B. Histogram Formation and Feature Dimensionality Reduction

After analyzing the given image using LTrP as outlined in the previous section, the 13 binary patterns (12 binary direction patterns + 1 magnitude pattern) are utilized to construct the histogram of the image using

$$H_k(l) = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} f_4(LP_k(i,j), l) \tag{6}$$

where, $l \in (0 \ldots 255)$, LP_k is the local pattern with $k \in (1, 2, \ldots, 13)$, $M \times N$ is the image resolution and

$$f_4(x,y) = \begin{cases} 1, & \text{if } x = y \\ 0, & \text{elsewhere} \end{cases} \tag{7}$$

The histograms corresponding to the 13 binary patterns are analyzed using the Kullback-Leibler (KL) divergence [16] with the motivation of reducing its dimensionality. The KL divergence between two histograms H_{k_1} and H_{k_2} is estimated using

$$D_{k_1,k_2}(H_{k_1}||H_{k_2}) = \sum_{i=1}^{r} H_{k_1}(i) log_b \left(\frac{H_{k_1}(i)}{H_{k_2}(i)} \right) \tag{8}$$

where, $(k_1, k_2) \in (1, 2, \ldots, 13]$, 'r' is no of bins (i.e 255) and 'b' is 10. Utilizing Eq. (8), the distance matrix D_{ij} is formed having dimension 169 (13×13). Since KL divergence is asymmetric, the J-divergence [17] is utilized to reduce the feature dimension i.e.

$$JD(H_{k_1}||H_{k_2}) = D_{KL}(H_{k_1}||H_{k_2}) + D_{KL}(H_{k_2}||H_{k_1}) \tag{9}$$

Eq. (9) reduces the dimensionality from 169 to 78 i.e. $\left(\frac{13 \times 13 - 13}{2}\right)$ thereby forming a 78-D feature vector at the end of this stage. The 78-D feature vector is now fed to the SVM classifier to obtain the class labels.

IV. Results and Discussions

MATLAB 2015a is used for carrying out the experimentation. UCID v2 [14] dataset is utilized for carrying out the experimentation. This dataset contains 1338 uncompressed TIFF color images which are firstly converted to gray scale. Next, the below listed datasets are formed:

i. Images of 3 sizes i.e., 64×64, 128×128 and 256×256, are cropped from the center of its full size source images (512×384). The dataset is denoted as (S^{ORI}).
ii. Median filtered datasets in which 3×3 and 5×5 median filtering was performed on S^{ORI} to obtain (S^{MF3}) and $S^{MF5})$, respectively.

Detailed comparative analysis of the proposed method is carried out with Zhang et al. method [12] which we have re-implemented and tested on the above dataset. The motivation for choosing Zhang et al. [12] method is two-fold. Firstly, it also uses a texture feature namely the local ternary patterns (LTP) for detecting median filtering. Secondly, their method improved earlier state-of-the-art techniques of median filtering detection namely SPAM [4] and MFF [18] thereby reducing our comparative analysis task. In addition, we also perform additional tests not carried out by Zhang et al. like differentiating between 3×3 and 5×5 median window sizes, detection and role of JPEG pre-compression as well as cut-paste forgery detection on composite images. In all our tests, we validate the proposed method's superiority in comparison to Zhang et al. [12]. To perform classification, we used C-SVM with RBF kernel. For each training-testing pair, 40% of median and 40% non-median samples are picked randomly to train the SVM classifier and the remaining samples are used for testing. Grid search along with five-fold cross validation on the training set is used to obtain optimum parameters (C, γ) in range $\{(2^i, 2^j) | i \in \{$ 0, 0.5,. . .10$\}$, $j \in \{$ -5,-4.5,....3 $\}\}$. The detection accuracy given below [12] is taken as the average of probability of true positive rate and true negative rate and this is averaged over 30 random experiments.

$$Accuracy(\%) = \left(\frac{P_{tp} + P_{tn}}{2} \right) \times 100 \qquad (10)$$

The classifier is retrained for each test so that the analysis carried out is robust and optimized. Comparative analysis is also carried out with one of the deep learning method proposed by Chen et al. [9] in-order to demonstrate the pros and cons of both methods. Results reported in the paper are reproducible and MATLAB code is available at https://sites.google.com/site/manishokade/publications.

A. Detection of median filtering in JPEG post-compressed images

The proposed method is tested to detect 3×3 median as well as 5×5 median in images that were JPEG post-compressed with varying quality factors ranging from 30 to 90. Table. I shows the detection accuracy achieved for Zhang et al. [12] method and proposed method. Few observations that can be made from the table include, higher accuracy values being achieved for the proposed method due to the streaking fingerprints being detected effectively by the application of local tetra patterns, larger image sizes leading to better detection performance since sufficient statistics being available, larger window sizes of the median filter achieving better accuracy due to longer streaking pixels being present and finally reduced JPEG quality factors achieving lower accuracy values due to the quantization step of the JPEG pipeline being larger thereby erasing the streaking fingerprints.

B. Detection of median filtering on JPEG pre-compressed images

In this section we study the effect of JPEG pre-compression on the performance of proposed method. 1338 uncompressed images were firstly JPEG compressed with quality factor 90, next 5×5 median filtering was carried out on these 1338 images and finally they were JPEG compressed with quality factor 70. This set is denoted by $JPEG - 90 + MF - 5 + JPEG - 70$. Another set consisting of all 1338 images from S^{ORI} was chosen where JPEG compression was carried out with quality factor 70 and no median filtering was applied on these images. This set is denoted by $JPEG - 70$. 400 samples (200 from each set) were chosen for testing. These samples were fed to the proposed median filtering detection scheme which gave the class labels. Utilizing the estimated class labels from the proposed method and the true class labels, confusion matrix was obtained. Same methodology was repeated for Zhang et al. method. Table. II(a)-(b) shows the confusion matrix obtained for Zhang et al. [12] method and proposed method, respectively. The results demonstrate lower misclassification rate for the proposed method in comparison to Zhang et al. method indicating JPEG pre-compression has less effect on the detection performance of the proposed method.

C. Distinguishing median filter window sizes

On detecting that median filtering has been carried out, there may be a requirement in some forensic application to determine the window size used to carry out the median filtering. The proposed method is able to carry out this task effectively by distinguishing 3×3 from 5×5 window size of the median filter. The S^{MF3} and S^{MF5} dataset is used for this purpose which consists of 1338 images that have been median filtered using 3×3 window size and 1338 images that have been median filtered using 5×5 window size thereby forming a total of 2676 images for experimental validation. The 2676 images are next JPEG compressed with quality factor of 70. This forms the dataset $S^{MF-3-5} + JPEG - 70$. The proposed method as well as Zhang et al. [12] method is used to distinguish between the two window sizes. Table III(a)-(b) shows the confusion matrix obtained for both methods when a total of 400 samples (200 samples from each class) are utilized for testing. As observed, the proposed method has a lower probability of misclassification in comparison to Zhang et al. [12] method thereby validating the superior performance of the proposed application of LTrP for median filtering window size detection.

D. Detection of cut-paste forgery

To demonstrate the performance of the proposed detection method, we apply 3×3 median filter on a source image and cut a certain portion from this image and paste it on an unaltered

TABLE I: Detection accuracy incase of JPEG post- compression for varying quality factors

Quality factor	64 × 64				128 × 128				256 × 256			
	MF3		MF5		MF3		MF5		MF3		MF5	
	Zhang [12]	Proposed	Zhang [12]	Proposed	Zhang [12]	Proposed	Zhang [12]	Proposed	Zhang [12]	Proposed	Zhang [12]	Proposed
JPEG 30	64.75	70.27	68.30	74.91	72.69	80.25	77.16	83.08	74.72	80.22	77.30	83.72
JPEG 40	65.81	71.47	69.50	76.52	73.36	81.36	78.44	83.21	75.27	82.50	82.52	87.30
JPEG 50	65.91	72.83	73.36	80.00	74.36	81.75	79.44	83.41	77.63	83.41	86.40	88.27
JPEG 60	68.94	75.33	74.05	80.63	76.80	84.38	79.77	83.80	78.16	85.25	86.66	88.80
JPEG 70	69.52	76.86	75.58	81.41	77.11	85.13	79.97	84.28	81.08	86.13	87.41	88.86
JPEG 80	72.55	77.72	76.72	81.97	78.19	85.88	80.27	85.97	83.52	86.41	87.56	89.50
JPEG 90	76.11	83.38	81.33	85.86	82.55	86.77	83.50	87.50	84.61	87.80	87.86	90.25

TABLE II: Confusion matrix for JPEG pre-compressed images

(a) Zhang et al. [12] method

		Estimated Label		
		JPEG90+MF5+JPEG70	JPEG70	Total
True Label	JPEG90+MF5+JPEG70	153	47	200
	JPEG70	36	164	200
	Total	189	211	400

(b) Proposed method

		Estimated Label		
		JPEG90+MF5+JPEG70	JPEG70	Total
True Label	JPEG90+MF5+JPEG70	172	28	200
	JPEG70	33	167	200
	Total	205	195	400

TABLE III: Confusion matrix for differentiating between 5×5 median and 3×3 median window sizes

(a) Zhang et al. [12] method

		Estimated Label		
		5 median	3 median	Total
True Label	5 median	165	35	200
	3 median	87	113	200
	Total	252	148	400

(b) Proposed method

		Estimated Label		
		5 median	3 median	Total
True Label	5 median	173	27	200
	3 median	24	176	200
	Total	197	203	400

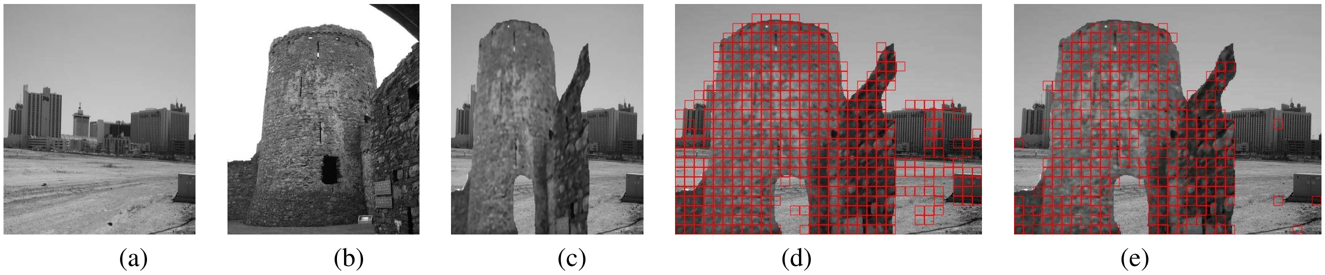

Fig. 3: (a) Unaltered image (b) Median filtered image (c) Composite image (d) Zhang et al. method [12] (e) proposed method.

image. Fig. 3(b) shows the image on which 3×3 median filter is applied on the entire image and a certain portion around the fort is cut followed by pasting it on an unaltered image shown in Fig. 3(a). The composite image is shown in Fig. 3(c) and this is saved with JPEG quality factor of 70. Now, the composite compressed image is analyzed by segmenting it into 16×16 fixed sized pixel blocks followed by testing each block for presence of median filtering. Fig. 3(d) and Fig. 3(e) shows the detection results obtained by Zhang et al. [12] method and proposed method, respectively with the red outline used to show the median filtering detections. As observed, Zhang et al. [12] method has many false positives like the region around the buildings and the ground being identified as median filtered in comparison to the proposed method. The better detection performance for the proposed method is due to its ability in identifying streaking pixels in comparison to Zhang's method.

TABLE IV: Average running time in seconds

Method	Zhang et al. [12]			Proposed		
Image size	64×64	128×128	256×256	64×64	128×128	256×256
Feature extraction	287.29	1127.51	4620.15	184.23	663.81	2812.63
Dimension reduction	5.28	5.43	5.41	0.04	0.04	0.05
Train/Test SVM	1.71	1.40	1.30	0.56	0.50	0.58
Total time	294.28	1134.34	4626.86	184.83	664.35	2813.26

E. Computational complexity analysis

Intel Core i7-4770 CPU running @ 3.4 GHz with 8 GB RAM is used for the simulation studies. Table IV shows the average processing times obtained for both methods. As observed, the proposed method achieves lower processing time for all image sizes i.e 64×64, 128×128 and 256×256. The lower computational complexity achieved is due to the three factors listed in Table IV namely feature extraction time, dimension reduction time and training/testing time. Firstly, the feature extraction time is less with LTrP in comparison to Zhang et al. [12] since in their method the local ternary pattern (LTP) is combined with local derivative pattern (LDP) to form the 2nd order LTP which takes more computational time. Secondly, we utilize the J-Divergence for feature dimensionality reduction while in their method KPCA is used which consumes more time. Finally, the proposed method has a 78-D feature vector in comparison to the 220-D feature vector in case of Zhang et al. [12] which reduces the training/testing time for the SVM classifier.

F. Performance analysis with a deep learning method

The final section highlights the trade-off offered by traditional methods and one of the deep learning based method investigated by Chen et al. [9]. The image size used in their work is 64×64, median filtering window is 3×3, compression QF is set to 70. Chen's method extracts Median Filtering Residual (MFR) and feeds this to the filter layer which then goes to the set of CNN's. They report that extracting MFR improves accuracy as opposed to directly feeding the pixels. Table V shows the accuracy obtained under the above setting by Chen's method in comparison to the proposed method. As observed their method outperforms the proposed method by 10 % for MFR case while the performance of their method is marginally better than the proposed method in Non-MFR case.

TABLE V: Detection accuracy incase of JPEG post- compression for 64×64 image size

Quality Factor	Chen et al. method [9]		Proposed method
	with MFR	without MFR	
JPEG-70	85.4	77.92	76.86

TABLE VI: Average running time in minutes

Method	Total Time
Chen et al.	15.12
Proposed	4.90

Running time analysis is carried out and results obtained are reported in Table VI. It can be observed that Chen's method is computationally complex in comparison to proposed method although it achieves 10 % better accuracy as reported in Table V. This is on expected lines since deep learners are time intensive but achieve better accuracy in comparison to hand crafted methods. However, it can be noted that they do not achieve very high accuracy in-spite of using CNN signifying that hand-crafting may be best suited for median filtering detection. Moreover their method cannot be termed fully deep since they are extracting MFR at the first stage. Also, the response to adversarial input is not accounted.

V. Conclusions

This paper investigated the application of local tetra patterns along with J-divergence to the median filtering detection under the forensic umbrella of verifying authenticity. Experimental testing along with comparative analysis with two state-of-the-art methods under challenging environments like JPEG post compression, median window size detection, cut-paste forgery detection showed good detection performance for the proposed method. Our future work is to analyze the response of median filtering to adversarial inputs for the deep learning methods and establish its findings against our proposed method.

References

[1] 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, Dec 2013.

[2] G. Cao, Y. Zhao, R. Ni, L. Yu, and H. Tian, "Forensic detection of median filtering in digital images," in *IEEE International Conference on Multimedia and Expo*, July 2010, pp. 89–94.

[3] Y. Niu, Y. Zhao, and R. Ni, "Robust median filtering detection based on local difference descriptor," *Signal Processing: Image Communication*, vol. 53, pp. 65 – 72, 2017.

[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.

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

[6] H. Tang, R. Ni, Y. Zhao, and X. Li, "Median filtering detection of small-size image based on cnn," *Journal of Visual Communication and Image Representation*, vol. 51, pp. 162 – 168, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S104732031830018X

[7] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Learning rich features for image manipulation detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 1053–1061.

[8] B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, ser. IH&#38;MMSec '16. New York, NY, USA: ACM, 2016, pp. 5–10. [Online]. Available: http://doi.acm.org/10.1145/2909827.2930786

[9] J. Chen, X. Kang, Y. Liu, and Z. J. Wang, "Median filtering forensics based on convolutional neural networks," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1849–1853, Nov 2015.

[10] L. Yu, Y. Zhang, H. Han, L. Zhang, and F. Wu, "Robust median filtering forensics by cnn-based multiple residuals learning," *IEEE Access*, vol. 7, pp. 120 594–120 602, 2019.

[11] W. Fan, K. Wang, F. Cayre, and Z. Xiong, "Median filtered image quality enhancement and anti-forensics via variational deconvolution," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 5, pp. 1076–1091, May 2015.

[12] 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, March 2014.

[13] S. Murala, R. P. Maheshwari, and R. Balasubramanian, "Local tetra patterns: A new feature descriptor for content-based image retrieval," *IEEE Transactions on Image Processing*, vol. 21, no. 5, pp. 2874–2886, May 2012.

[14] G. Schaefer and M. Stich, "UCID: an uncompressed color image database," in *Proc. SPIE*, vol. 5307, 2003, pp. 472–480.

[15] A. Bovik, "Streaking in median filtered images," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 4, pp. 493–503, Apr 1987.

[16] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 03 1951.

[17] J. Lin, "Divergence measures based on the shannon entropy," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 145–151, Jan 1991.

[18] H. D. Yuan, "Blind forensics of median filtering in digital images," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 4, pp. 1335–1345, Dec 2011.