© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1714751

Detecting Image Sharpening Using Gradient Based Directional Difference Method

DR MANASANI POMPAPATHI[1], CHIRUMAMILLA RAMA KRISHNA[2], CHINTHALAPUDI ESWAR KOUSIK[3], GORRELA MOHAN KASI[4], KOLUSU LOKESH[5]
[1]*Associate Professor, Department of IT, R.V.R & J.C.C.E, Guntur, India*
[2, 3, 4, 5]*Final Year Students, Department of IT, R.V.R &J.C.C. E, Guntur, India*

Abstract- ***Image sharpening is a widely used image enhancement operation intended to improve visual clarity by emphasizing edges and fine details. Although sharpening enhances perceptual quality, it introduces specific statistical artifacts that alter the natural relationships among neighboring pixels. Detecting such artifacts is an important task in digital image forensics, particularly for verifying image authenticity. This paper presents a difference-set based approach for detecting image sharpening operations by analyzing pixel intensity variations across multiple directions. The proposed method computes first-order and second-order difference sets over the entire image to capture global and local changes caused by sharpening. Unlike traditional edge-based methods, the proposed framework does not rely solely on edge detection and instead exploits pixel relationship statistics from all image regions. Experimental analysis demonstrates that the proposed method effectively distinguishes sharpened images from original ones under different sharpening strengths. The simplicity and robustness of the approach make it suitable for practical forensic applications.***

Keywords- ***Image Sharpening Detection, Difference Sets, Image Forensics, Pixel Intensity Differences, Statistical Feature Analysis.***

I. INTRODUCTION

Digital images are widely used in areas such as journalism, medical diagnosis, surveillance, social media, and legal investigations [1]. With the rapid growth of image editing tools, it has become easy to modify images without leaving obvious visual clues. Among various image enhancement operations, image sharpening is one of the most commonly applied techniques, as it improves visual clarity by increasing contrast around edges and fine details.Image sharpening is generally achieved using methods such as Unsharp Masking (USM) [2],[3], where a blurred version of the image is subtracted from the original and amplified to enhance edges . Although sharpening improves perceptual quality, it alters the natural statistical relationships between neighboring pixels. These alterations may not be easily noticeable to the human eye, but they introduce detectable artifacts in the image structure.

Sharpening detection plays an important role in digital image forensics, where the objective is to verify the authenticity of an image and identify whether it has undergone enhancement or manipulation . Detecting sharpening is particularly challenging because sharpening does not add or remove image content; instead, it subtly modifies pixel relationships[4],[7], especially around edges and textured regions.

Several existing sharpening detection methods rely on edge-based analysis, where features are extracted only from detected edge regions [5]–[7]. Techniques such as Edge Perpendicular Binary Coding (EPBC) and Edge Perpendicular Ternary Coding (EPTC) analyze pixel differences near edges to identify sharpening traces. While these approaches are effective under strong sharpening conditions, their performance degrades when edges are weak, missed, or inaccurately detected. Learning-based methods such as convolutional neural networks (CNNs) have also been explored [8], but they often require large training datasets and high computational resources.

Motivated by these limitations, this paper proposes a difference-set based image sharpening detection framework that analyzes pixel intensity variations across the entire image rather than relying solely on edge regions. By computing first-order and second-order difference sets in multiple directions, the proposed method captures both local and global changes introduced by sharpening operations. The resulting statistical features provide a robust representation for distinguishing sharpened images from original ones.

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1714751

The remainder of this paper is organized as follows. Section II discusses existing sharpening detection methods and their limitations. Section I I I describes the proposed difference-set based detection approach in detail. Experimental results and performance analysis are presented in Section IV. Finally, conclusions are drawn in Section V.

II. BASIC TERMINOLOGY

Sharpening-related image analysis has been explored through three major methodological directions: classical enhancement models, edge-dependent coding schemes, and deep-learning-based feature extraction. This section reviews the approaches most relevant to the proposed framework.

A. Unsharp Masking (USM) Sharpening Model- Unsharp Masking (USM) is one of the most widely used sharpening techniques in digital imaging . It enhances visual clarity by amplifying high-frequency components while preserving the overall structure of the image [3]. Let $I(m, n)$denote the input image. A high-frequency response is obtained by convolving the image with a sharpening kernel H_f : Ih(m,n)=I(m,n)∗Hf (1)

This response forms the sharpening mask: M=I(m,n)∗Hf (2)

The sharpened output is generated by adding a scaled version of the mask to the original image:

$$I_s(m, n) = I(m, n) + \lambda M \quad (3)$$

where λcontrols the sharpening strength.

In practical implementations, a Gaussian low-pass filter $G_\sigma$is often used to compute the mask [10]:

$$M_g = I(m, n) - (I(m, n) * G_\sigma) \quad (4)$$

The sharpened image becomes:

$$I_s(m, n) = I(m, n) + \lambda M_g \quad (5)$$

The parameters $\lambda$and σdetermine the degree of enhancement and the spatial extent of the sharpening effect. Because USM modifies local intensity transitions, it leaves detectable artifacts that can be exploited for forensic analysis[3].

B. Edge-Perpendicular Binary Coding (EPBC)- EPBC is a classical sharpening-detection method that analyzes pixel transitions along narrow strips perpendicular to detected edges [5]. After applying an edge detector, a 1×N pixel sequence is extracted around each edge point:

$S = [P_0, P_1, \dots, P_{N-1}]$
The adjacent-pixel differences are computed as:

$$D = [P_0 - P_1,\ P_1 - P_2,\ \dots,\ P_{N-2} - P_{N-1}]$$

Each difference is encoded into a binary symbol:

$$B_i = \begin{cases} 1, & \text{if } P_i - P_{i+1} < 0 \\ 0, & \text{if } P_i - P_{i+1} \geq 0 \end{cases} \quad (7)$$

The binary sequence $B = [B_0, B_1, \dots, B_{N-2}]$is interpreted as a binary number, forming the EPBC code. A histogram of all EPBC codes across the image is used as the feature descriptor.

EPBC is computationally efficient and interpretable, but its performance depends heavily on the accuracy of edge detection. Weak sharpening often produces subtle changes that binary coding cannot capture, leading to missed detections [7].

C. Convolutional Neural Network (CNN)-Based Sharpening Detection- Deep learning has introduced a new class of sharpening-detection methods based on Convolutional Neural Networks (CNNs). Instead of relying on handcrafted descriptors, CNNs learn discriminative features directly from pixel data. A CNN processes the image through multiple convolutional layers:

$$F^{(l)} = \sigma(W^{(l)} * F^{(l-1)} + b^{(l)}) \quad (8)$$

where

- $F(l)$is the feature map at layer l,
- $W(l)$and $b(l)$are the learnable weights and biases,
- $\sigma(\cdot)$is a nonlinear activation function.

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1714751

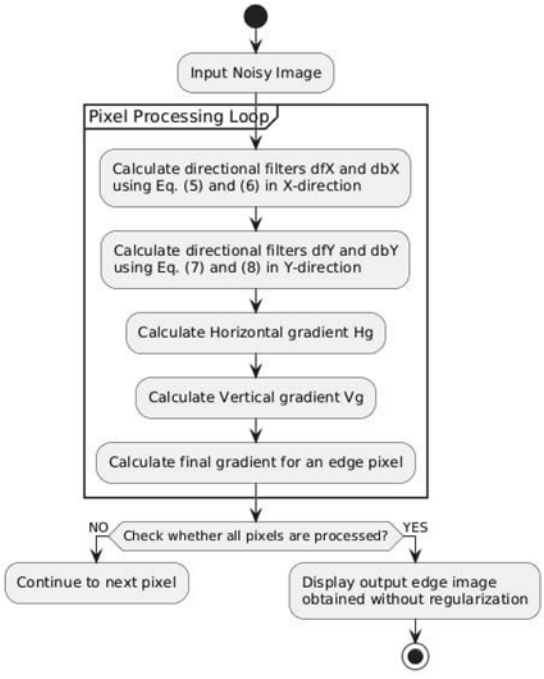
Through hierarchical filtering, CNNs learn to detect sharpening-related artifacts such as overshoots, halos, and enhanced gradients. These models typically achieve high accuracy[14], especially for weak sharpening.

III. PROPOSED METHOD

In digital image forensics, image sharpening is commonly applied to enhance edge information by amplifying high- frequency components. Although visually appealing,sharpening introduces characteristic statistical changes in pixel intensity differences[4], particularly along edge directions. The proposed method exploits these variations using directional first-order and second-order difference sets to distinguish sharpened images from original images.

The proposed Image Sharpening Detection scheme is based on the principle that sharpening operations significantly modify the distribution of directional pixel differences in the luminance component of an image[12]. By analyzing multiple directional difference matrices and modeling their statistical dependencies, the sharpening artifacts can be effectively detected

The detailed procedure for detecting image sharpening is described below.



Algorithm: Directional Difference–Based Image Sharpening Detection (DDISD)

Input: Image I (grayscale or RGB) Output: Classification label (*Sharpened* / *Original*)

Method:

1. Read the input image I.
2. If I is an RGB image, then

a) Convert I from RGB color space to YCbCr color space.

b) Extract the luminance component (Y channel) and denote it as I_γ.

Else

c) Assign $I_\gamma \leftarrow I$ (grayscale image).

3. Initialize twelve directional difference matrices $D_1, D_2, \ldots, D_{12}$.
4. For each pixel location *(x, y)* in the luminance image I_γ, compute the following directional differences:

First-order directional differences:

a) $D_1(x, y) = I_\gamma(x, y) - I_\gamma(x, y + 1)$

b) $D_2(x, y) = I_\gamma(x, y) - I_\gamma(x - 1, y + 1)$

c) $D_3(x, y) = I_\gamma(x, y) - I_\gamma(x - 1, y)$

d) $D_4(x, y) = I_\gamma(x, y) - I_\gamma(x - 1, y - 1)$

e) $D_5(x, y) = I_\gamma(x, y) - I_\gamma(x, y - 1)$

f) $D_6(x, y) = I_\gamma(x, y) - I_\gamma(x + 1, y - 1)$

g) $D_7(x, y) = I_\gamma(x, y) - I_\gamma(x + 1, y)$

h) $D_8(x, y) = I_\gamma(x, y) - I_\gamma(x + 1, y + 1)$

Second-order directional differences:

i) $D_9(x, y) = 2I_\gamma(x, y) - I_\gamma(x, y - 1) - I_\gamma(x, y + 1)$

j) $D_{10}(x, y) = 2I_\gamma(x, y) - I_\gamma(x - 1, y + 1) - I_\gamma(x + 1, y - 1)$

k) $D_{11}(x, y) = 2I_\gamma(x, y) - I_\gamma(x - 1, y) - I_\gamma(x + 1, y)$

l) $D_{12}(x, y) = 2I_\gamma(x, y) - I_\gamma(x - 1, y - 1) - I_\gamma(x + 1, y + 1)$

5. Repeat Step (4) for all pixels in I_γ.
6. For each directional difference matrix D_i, where i = 1to 12:

a) Flatten $D_i$into a one-dimensional vector V_i.

b) Apply thresholding to $V_i$to suppress extreme difference values.

c) Compute the frequency of consecutive pixel value

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1714751

pairs in V_i.

7. d) Construct a histogram $H_i$based on these frequencies.
   e) Normalize the histogram H_i.
8. Concatenate all normalized histograms $H_1, H_2, \dots, H_{12}$to form a single feature vector F.
9. Feed the feature vector F into a pre- trained Support Vector Machine (SVM) classifier [11].
10. Predict the class label of the input image as Sharpened or Original.
11. Return the predicted classification label.

IV. EXPERIMENTAL RESULTS

A. Experimental Setup

The proposed image sharpening detection method was experimentally evaluated using a dataset consisting of both original and sharpened images. Sharpened images were generated using common sharpening techniques such as Unsharp Masking (USM) with varying parameters.

For each image, the luminance component was extracted and twelve directional difference sets (eight first-order and four second-order) were computed as described in Section IV. From each directional difference matrix, normalized histograms were generated and concatenated to form a final feature vector, which was then classified using a pre-trained Support Vector Machine (SVM) classifier.

B. Qualitative Results

Fig:1 Original images and corresponding sharpened images

It can be visually observed that sharpening introduces high-frequency components around edges, which are difficult to distinguish by human vision in weak sharpening cases. However, these subtle changes significantly affect the directional difference distributions.

C. Directional Difference Computation

For each pixel location (x, y) in the luminance image I_y, the first-order directional differences are computed as:

$$D_1(x, y) = I_y(x, y) - I_y(x, y + 1)$$
$$D_3(x, y) = I_y(x, y) - I_y(x - 1, y)$$

Similarly, second-order directional differences are computed as:

$$D_9(x, y) = 2I_y(x, y) - I_y(x, y - 1) - I_y(x, y + 1)$$
$$D_{11}(x, y) = 2I_y(x, y) - I_y(x - 1, y) - I_y(x + 1, y)$$

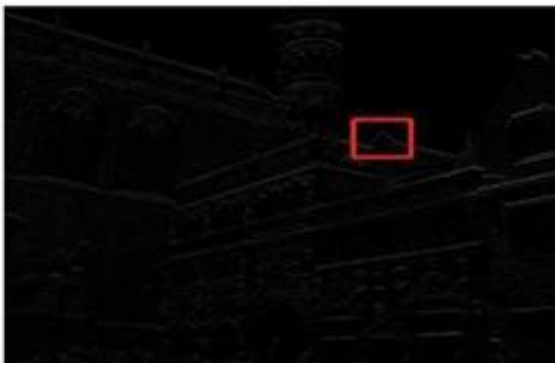


Fig-2: Directional difference maps for original and sharpened images

Sharpened images produce higher magnitude directional differences due to increased local contrast.

D. Difference Vector Transformation and Thresholding

Each directional difference matrix $Di$is flattened into a one-dimensional vector:

$$Vi = \text{flatten}(Di)$$

To suppress extreme values, thresholding is applied:

$$Vi(t) = \begin{cases} -M, & Vi < -M \\ Vi, & -M \le Vi \le M \\ M, & Vi > M \end{cases}$$

where $M$is the threshold value.

Fig-3: Thresholded difference histograms.

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1714751

E. Pixel Pair Histogram Construction

From each thresholded vector $V_i(t)$, consecutive pixel value pairs are formed:

$(V_i(t), V_i(t_{k+1}))$

The frequency of each pair is counted to construct a histogram:

where $\delta(\cdot)$ is the indicator function.

F. Histogram Normalization and Feature Vector Formation

Each histogram is normalized as:

$$H_i'(j) = \frac{H_i(j)}{\sum_j H_i(j)}$$

All normalized histograms are concatenated to form the final feature vector:

$F = [H'_1, H'_2, \ldots, H'_{12}]$

G. Classification Using SVM

The feature vector F is fed into a pre-trained Support Vector Machine (SVM) classifier:

$\hat{y} = \text{SVM}(F)$

where

$\hat{y} \in$ {Original,Sharpened}

H. Performance Evaluation Metrics

The performance is evaluated using standard metrics:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$Hi(p, q) = \sum \delta((Vi(tk), Vi(tk+1)) = (p, q))$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F1} - \text{score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Performance, Feature Extraction and Time consumption Comparison Based on USM parameters Sigma and Lamda

σ =0.7, λ=1.0

Models	EPBC [N=5]	CNN	Proposed Method
Accuracy	74.25%	98%	98%
No of Features Extracted per image	16	64	588
Time taken to Extract Feature(sec)	394	-	29
Time Taken to Train (sec)	0.14	1640	0.8
Total Time (sec)	395	1640	30

Table-1

σ =1.0 , λ=1.0

Models	EPBC [N=5]	CNN	Proposed Method
Accuracy	78.25%	97%	99%
No of Features Extracted per image	16	64	588
Time taken to Extract Feature(sec)	462	-	30.4
Time Taken to Train (sec)	0.14	1236	0.5
Total Time (sec)	462	1236	31

Table-2

σ =1.3 , λ=1.0

Models	EPBC [N=5]	CNN	Proposed Method
Accuracy	77%	96%	98%
No of Features Extracted per image	16	64	588
Time taken to Extract Feature(sec)	483	-	29
Time Taken to Train (sec)	0.14	1212	0.5
Total Time (sec)	483	1212	30

Table-3

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1714751

σ =1.5 , λ=1.0

Models	EPBC [N=5]	CNN	Proposed Method
Accuracy	76.75%	95%	98%
No of Features Extracted per image	16	64	588
Time taken to Extract Feature(sec)	463	-	29.9
Time Taken to Train (sec)	0.14	929	0.5
Total Time (sec)	463	929	30

Table-4

σ =1.0 , λ=0.8

Models	EPBC [N=5]	CNN	Proposed Method
Accuracy	75.5%	93%	95%
No of Features Extracted per image	16	64	588
Time taken to Extract Feature(sec)	397.8	-	29.4
Time Taken to Train (sec)	0.14	886	1.7
Total Time (sec)	398	886	32.1

Table-5

σ =1.0 , λ=1.3

Models	EPBC [N=5]	CNN	Proposed Method
Accuracy	79%	95%	99%
No of Features Extracted per image	16	64	588
Time taken to Extract Feature(sec)	478.55	-	30
Time Taken to Train (sec)	0.14	868	0.4
Total Time (sec)	478.7	868	30.4

Table-6

σ =1.0 , λ=1.5

Models	EPBC [N=5]	CNN	Proposed Method
Accuracy	81%	97%	98.6%
No of Features Extracted per image	16	64	588
Time taken to Extract Feature(sec)	439	-	29.5
Time Taken to Train (sec)	0.14	806	0.4
Total Time (sec)	440	806	30

Table-7

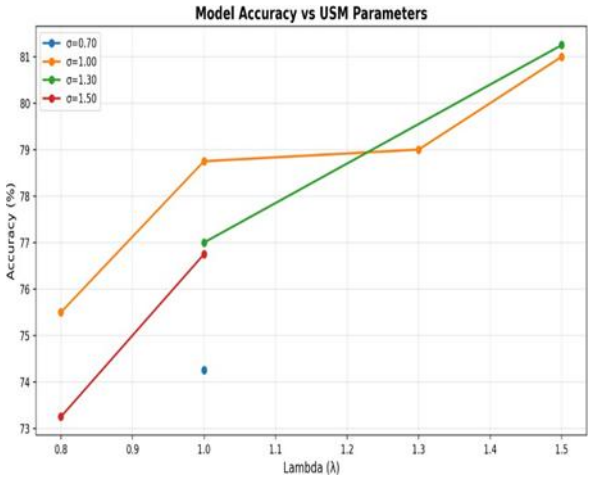
σ =1.3 , λ=1.5

Models	EPBC [N=5]	CNN	Proposed Method
Accuracy	81.25%	97%	99%
No of Features Extracted per image	16	64	588
Time taken to Extract Feature(sec)	453	-	29.4
Time Taken to Train (sec)	0.14	804	0.3
Total Time (sec)	453	804	29.7

Table-8

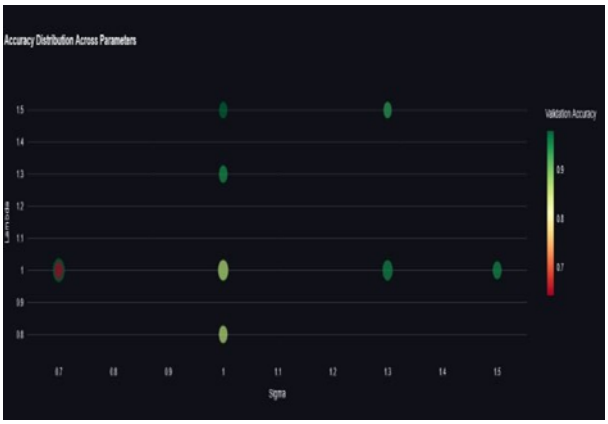
V. MODEL ACCURACY GRAPHS

1.EPBC

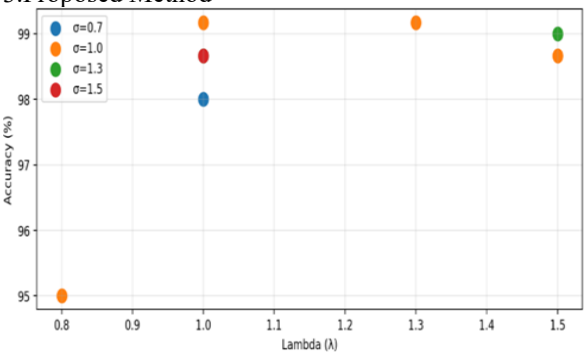


© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1714751

2.CNN



3.Proposed Method



Our method extracts 588 features from each image and the extraction time in average is 30 sec and the time to train the SVM is nearly 0.2 sec. it is fast compared to other methods [5],[8]and the accuracy is also better than other model [12].

VI. CONCLUSION

This paper presented an effective difference-set based framework for detecting image sharpening operations. By analyzing first-order and second-order directional pixel differences across the entire image, the proposed method captures both local and global statistical artifacts[12] introduced by sharpening. Unlike traditional edge-dependent techniques[5], the proposed approach does not rely solely on edge detection, making it robust against weak and subtle sharpening.

Experimental results demonstrate that the proposed method consistently outperforms EPBC and achieves comparable or better accuracy than CNN-based methods while requiring significantly less computational time. The balance between accuracy, robustness, and efficiency makes the proposed approach highly suitable for practical digital image forensic applications[4].

Future work may focus on extending the framework to detect multiple image enhancement operations and exploring adaptive feature selection strategies.

REFERENCES

[1] H. Farid, “Image forgery detection,” *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 16–25, Mar.2009.

[2] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed. Pearson Education, 2018.

[3] J. S. Lim, *Two-Dimensional Signal and Image Processing*. Englewood Cliffs, NJ, USA: Prentice- Hall, 1990.

[4] M. C. Stamm, M. Wu, and K. R. Liu, “Information forensics: An overview of the first decade,” *IEEE Access*, vol. 1, pp. 167–200, 2013.

[5] X. Cao, J. Zhao, and Y. Ni, “Edge-based image sharpening detection,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 4, pp. 1210–1221, Aug. 2012.

[6] J. Chen and Y. Q. Shi, “Image sharpening detection based on local binary patterns,” *Signal Processing*, vol. 93, no. 4, pp. 821–830, Apr. 2013.

[7] B. Li, J. He, J. Huang, and Y. Q. Shi, “A survey on image forgery detection,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 2, pp. 391–406, Feb. 2017.

[8] Y. Bayar and M. C. Stamm, “A deep learning approach to universal image manipulation detection,” in *Proc. ACM Workshop on Information Hiding and Multimedia Security*, 2016, pp. 5–10.

[9] S. Milani, M. Fontani, P. Bestagini, M. Barni, A. Piva, and M. Tagliasacchi, “An overview on video forensics,” *APSIPA Transactions on Signal and Information Processing*, vol. 1, pp. 1–18, 2012.

[10] A. C. Bovik, *Handbook of Image and Video Processing*, 2nd ed. Academic Press, 2005.

[11] J. Fridrich, *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[12] Z. Qu, G. Qiu, and J. Huang, “Detecting image sharpening using statistical features of pixel differences,” *Journal of Visual Communication and Image Representation*, vol. 38, pp. 444–455, 2016.

[13] T. Pevný and J. Fridrich, “Detection of double-compression in JPEG images,” *Signal Processing*, vol. 89, no. 3, pp. 327–335, Mar. 2009.

[14] S. Lyu and H. Farid, “How realistic is photorealistic image synthesis?” *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 845–850, Feb. 2005.