Close-to-optimal Counter Histogram-Based Forensics using Mean Structural Similarity Index Metric

Reza Kazemi, Arash Amini,Borna Khodabandeh, Morteza Alikhani

***Abstract*—Image forensics and counter forensics (CF) are two competing fields that have experienced significant developments in recent years. Interestingly, the use of histogram is popular in both forensic detectors and counter-forensic methods. In this work, we focus on the histogram-based CF methods; in particular, we propose a quasi-convex version of SSIM and MSSIM as the cost function of CF which helps in restricting search domain for optimal solution to the CF problem. Also, we propose two sub-optimal methods for this problem: 1) a gradient descent version of the optimal counter-forensics Method (OCM) with the cost function MSSIM instead of MSE (which we call GDOCM), and 2) another method that employs unitary matrices as the transfer matrix (which we call UMM). We numerically compare the proposed methods with the OCM method in different settings including the common JPEG compression detection scenario. Our experiments confirm superiority of the proposed methods compared to OCM.**

***Index Terms*—Counter forensics, histogram-based forensics, image forensics, SSIM**

I. Introduction

With the spread of multimedia contents and image processing and generating tools, originality of images and image integrity have become important. The release of manipulated or fake images influences different areas such as judicial systems, economics, etc [1], e.g. [2]–[4]. As a result, digital image forensics has become an important and growing area of research. This caused a rivalry between forensics and counter forensics [5]. The goal in forensics is to determine whether a given content is original or deliberately manipulated. In contrast, the goal in counter-forensics (CF) techniques is to mislead forensics detectors.

Related Works

The CF methods can be categorized based on different aspects. [6] reviews various categories for image integrity techniques and [7] studies different categories of CF techniques based on their manipulation approaches and the image forensics detection methods they aim to attack. More recent CF and counter-CF methods are based on machine learning (ML); [1] reviews some of the image forensics, counter forensics, and anti-counter forensics. Deep learning (DL) is also strong tool employed for image forensics [8]. A limitation of the ML and DL-based methods in real applications is their vulnerabilities in the detection of adversarial and train-test data mismatch [1]. For this reason, traditional methods (even heuristic methods), are still of great importance. It is also possible to combine them with new ML-based methods.

Two of the widely used counter image forensics detectors are JPEG and double JPEG compression [9]–[12]. To bypass the double JPEG compression test, some improvements for the forensics methods are proposed [13]–[16]. Later, some other works in the field of double JPEG compression detectors in image forensics have been done such as in [13]–[16]. In particular, [16] proposed an adversary-aware double JPEG compression detector that detects universal counter forensics of double JPEG compression.

Analyzing the statistics of an image, including its histogram, is the most common image forensics technique. [17] introduced an algorithm to detect contrast enhancement based on the fact that natural images have smooth histograms in contrast to manipulated contrast-enhanced images that have some picks and gaps artifacts in their histogram. A universal CF approach that aims to hide traces on the histogram of the images has been introduced in [18], [19]. Similarly, optimal attacking strategies are proposed in [5], [20] for the histogram-based forensics detectors assuming the cost function is convex. [5] used mean-squared error (MSE) as its convex cost function and introduced optimal counter-forensics method (OCM) to find the best solution for the CF problem.

The downside of using MSE as the cost function is that it is not fully aligned with the image quality that humans perceive. Although the structural similarity index (SSIM) is a better metric, it is not convex. Therefore, both finding and implementing the optimal solution based on SSIM might be computationally impractical. In this work, instead, we introduce practical sub-optimal methods using the SSIM metric.

Deep learning has greatly enhanced image and video forensics [21]–[23], especially in detecting deepfakes [24]–[27], by identifying subtle patterns and artifacts. CF techniques, however, aim to evade these detectors. Unlike typical CF methods that use adversarial attacks [28]–[30], our approach focuses on histogram-based forensics. We employ a sub-optimal method to maximize the perceptual metric of MSSIM, ensuring modifications are less noticeable to human observers, rather than fooling specific forensic detectors.

Manuscript received June 22, 2024; revised xxxx ??, 20??.

R. Kazemi is with the Electronics Research Institute, Sharif University of Technology, Tehran, Iran. Email: reza.kazemi@sharif.edu.

A. Amini B. Khodabandeh and M. Alikhani are with the Electrical Engineering department, Sharif University of Technology, Tehran, Iran. Email: aamini@sharif.edu, borna710kh@gmail.com and mortezaalikhani20@gmail.com.

Contributions

In this work, we introduce quasi-convex versions of the SSIM and mean SSIM (MSSIM) metrics and incorporate them as the cost function in the CF problem. Next, we show how to considerably restrict the search domain for the optimal solution to the CF problem.Particularly, we propose a sub-optimal method that uses the OCM method but is based on the gradient descent approach; we call it the gradient descent OCM (GDOCM). As the method is histogram-based, it leaves the histogram of the image unchanged. We propose yet another sub-optimal method called the unitary matrices method (UMM), by applying unitary matrices as transfer matrices. We examine both methods numerically and show that they outperform the traditional OCM technique in various tests including the JPEG compression setup.

Notation

Throughout this paper, we use regular lowercase letters for deterministic and random scalar variables, and lowercase boldface letters for vectors. Matrices are represented by regular uppercase letters, whose entries shall be denoted by lowercase version of the same letter equipped with subscripts. The notation $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^H \mathbf{y}$ represents the inner product of the vectors $\mathbf{x}$ and $\mathbf{y}$, where $\cdot^H$ is the conjugate transpose operation. Further, $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$ stands for the ℓ_2 norm of the vector $\mathbf{x}$. We define $\mathbb{1}(statement)$ as the boolean function that takes the value 1 in case the input statement is valid, and value 0 otherwise. For a vector $\mathbf{x}$ of size N_p, $\mathcal{F}(\mathbf{x})_k$ represents the kth discrete Fourier coefficient defined by $\sum_{n=0}^{N_p-1} \mathrm{e}^{-\mathrm{j}\frac{kn}{N_p}} x_n$, where $k = 0, \ldots, N_p - 1$. Furthermore, if $\mathcal{L} \subseteq \{0, \ldots, N_p - 1\}$, $\mathbf{x}_\mathcal{L}$ defines the vector $\mathbf{x}$ restricted to the indices in $\mathcal{L}$, *i.e.*, $\mathbf{x}_\mathcal{L} = (x_k)_{k \in \mathcal{L}}$. Finally, we utilize $\overline{\mathbf{x}}$ to indicate the sample mean of $\mathbf{x}$ defined by $\overline{\mathbf{x}} = \frac{\sum_{i=0}^{N_p-1} x_i}{N_p}$. Similarly, $\overline{X}$ extends the definition to matrices to represent the average of all elements.

II. Background

There are several criteria to assess the amount of similarity between two images. The two main categories are perceptual indices and canonical metrics. Oftentimes, the perceptual approaches are visually superior; however, due to their complicated structure, it is usually difficult to analyze them mathematically and incorporate them into optimization tasks for deriving image processing tools. The structural similarity (SSIM) index is among the well-known perceptual quality assessment metrics. It is defined as [31]:

$$\begin{aligned} \mathrm{SSIM}(\mathbf{x}', \mathbf{x}) &= S_1(\mathbf{x}', \mathbf{x}) \cdot S_2(\mathbf{x}', \mathbf{x}); \\ S_1(\mathbf{x}', \mathbf{x}) &= \frac{2\overline{\mathbf{x}}\overline{\mathbf{x}'} + c_1}{\overline{\mathbf{x}}^2 + \overline{\mathbf{x}'}^2 + c_1}, \\ S_2(\mathbf{x}', \mathbf{x}) &= \frac{2\sigma_{\mathbf{x}\mathbf{x}'} + c_2}{\sigma_\mathbf{x}^2 + \sigma_{\mathbf{x}'}^2 + c_2}, \end{aligned} \tag{1}$$

where c_1, c_2 are non-negative constants, $\mathbf{x}, \mathbf{x}'$ are the two images in vectorial form, $\sigma_\mathbf{x}^2, \sigma_{\mathbf{x}'}^2$ represent the sample variance of $\mathbf{x}$ and $\mathbf{x}'$ respectively, and $\sigma_{\mathbf{x}\mathbf{x}'}$ stands for the sample covariance between $\mathbf{x}$ and $\mathbf{x}'$. The latter quantities amount to

$$\sigma_\mathbf{x}^2 = \overline{\mathbf{x}^2} - \overline{\mathbf{x}}^2, \qquad \sigma_{\mathbf{x}\mathbf{x}'} = \frac{\sum_{i=0}^{N_p-1} (x_i - \overline{\mathbf{x}})(x_i' - \overline{\mathbf{x}'})}{N_p}, \tag{2}$$

where N_p is the length of $\mathbf{x}$ (or $\mathbf{x}'$). In this paper, c_1, c_2 are either assumed to be 0 or very small. It is important to mention that the SSIM is only an index and not a metric, as it does not satisfy the triangular inequality [31]. Therefore,

$$g^x(\mathbf{x}', \mathbf{x}) = \sqrt{1 - \mathrm{SSIM}(\mathbf{x}', \mathbf{x})} \tag{3}$$

is introduced in Section II-C of [31] as a modified version of the SSIM that defines a metric. Unfortunately, $g^x(., \mathbf{x})$ is not convex; nevertheless, it admits a low-order convex approximation.

Lemma 1. *For any* $\mathbf{x}$*, the low-order approximation of* $g^x(., \mathbf{x})$ *defined by*

$$\tilde{g}^x(\mathbf{x}', \mathbf{x}) \doteq \sqrt{2 - S_1(\mathbf{x}', \mathbf{x}) - S_2(\mathbf{x}', \mathbf{x})}, \tag{4}$$

is quasi-convex[1] *with respect to* $\mathbf{x}'$ *for all* $\mathbf{x}' \in \mathcal{Q}_\mathbf{x}$*, where*

$$\mathcal{Q}_\mathbf{x} = \left\{ \mathbf{z} \in \mathbb{R}^N \middle| \overline{\mathbf{z}} \in [0, \sqrt{3}\overline{\mathbf{x}}], \frac{\|(\mathbf{x} - \overline{\mathbf{x}}) - (\mathbf{z} - \overline{\mathbf{z}})\|}{\|(\mathbf{x} - \overline{\mathbf{x}})\|} \leq \sqrt{3} - 1 \right\}. \tag{5}$$

Proof. It is shown in Appendix A that

$$2 - S_1(\mathbf{x}', \mathbf{x}) - S_2(\mathbf{x}', \mathbf{x})$$

is a convex function of $\mathbf{x}'$ in the neighborhood $\mathcal{Q}_\mathbf{x}$ of $\mathbf{x}$. Thus, for all positive values of h

$$\mathcal{Q}_\mathbf{x} \cap \left\{ \mathbf{z} \;\middle|\; 2 - S_1(\mathbf{z}, \mathbf{x}) - S_2(\mathbf{z}, \mathbf{x}) \leqslant h \right\}$$

defines a convex set. Similarly,

$$\mathcal{Q}_\mathbf{x} \cap \left\{ \mathbf{z} \;\middle|\; \tilde{g}^x(\mathbf{z}, \mathbf{x}) \leqslant \sqrt{h} \right\}$$

is a convex set for all $h > 0$, which implies the quasi-convexity of $\tilde{g}^x(\mathbf{x}', \mathbf{x})$ restricted to $\mathbf{x}' \in \mathcal{Q}_\mathbf{x}$. ■

Remark 1. *The convexity region of* $2 - S_1(\mathbf{x}', \mathbf{x}) - S_2(\mathbf{x}', \mathbf{x})$ *- which itself contains* $\mathcal{Q}_\mathbf{x}$ *- is a subset of the quasi-convexity region for* $\tilde{g}^x$*. Nevertheless, the quasi-convexity region for* $\tilde{g}^x$ *might be larger than* $\mathcal{Q}_\mathbf{x}$ *in practice.*

The introduced SSIM criterion is usually considered as a basic similarity index between two windows rather than two images. The Mean SSIM (MSSIM) is one of the various ways to combine similarity indices of small windows inside two images to obtain an overall index as follows [32]:

$$\mathrm{MSSIM}(\mathbf{x}', \mathbf{x}) = \frac{1}{N_c} \sum_{i=1}^{N_c} \mathrm{SSIM}(\mathbf{x}'_{c_i}, \mathbf{x}_{c_i}), \tag{6}$$

where $\mathbf{x}_{c_i}$ denotes the vector $\mathbf{x}$ restricted to the pixels inside the ith local window and N_c represents the number of possible

[1] A quasi-convex function is a real-valued function defined on an interval or on a convex subset such that the inverse image of any set of the form $(-\infty, a)$ is a convex set, where a is an arbitrary real value.

local windows fitting inside the images. The windows are chosen in a sliding fashion which allows for considerable overlap. Extending the metric and quasi-convex approximations for the MSSIM, we arrive at

$$g^x_{_M}(\mathbf{x}',\mathbf{x}) \doteq \sqrt{1-\mathrm{MSSIM}(\mathbf{x}',\mathbf{x})},$$
$$\tilde{g}^x_{_M}(\mathbf{x}',\mathbf{x}) = \left(2-\frac{1}{N_c}\sum_{i=1}^{N_c} S_1\left(\mathbf{x}'_{c_i},\mathbf{x}_{c_i}\right)+S_2\left(\mathbf{x}'_{c_i},\mathbf{x}_{c_i}\right)\right)^{\frac{1}{2}}. \quad (7)$$

Lemma 2. *$\tilde{g}^x_{_M}(\mathbf{x}',\mathbf{x})$ is quasi-convex with respect to $\mathbf{x}'$ within the region*

$$\mathcal{Q}^M_{\mathbf{x}} = \bigcap_{i=1}^{N_c}\Big\{\mathbf{z}\in\mathbb{R}^N\Big|\overline{\mathbf{z}}_{c_i}\in[0,\sqrt{3}\overline{\mathbf{x}}_{c_i}], \frac{\|(\mathbf{x}_{c_i}-\overline{\mathbf{x}}_{c_i})-(\mathbf{z}_{c_i}-\overline{\mathbf{z}}_{c_i})\|}{\|(\mathbf{x}_{c_i}-\overline{\mathbf{x}}_{c_i})\|}\leq\sqrt{3}-1\Big\}. \quad (8)$$

Proof. Similar to $g^x(.,\mathbf{x})$, we can follow as

$$\begin{aligned}\tilde{g}^x_{_M}(\mathbf{x}',\mathbf{x})^2 &= 2-\frac{1}{N_c}\sum_{i=1}^{N_c} S_1(\mathbf{x}'_{c_i},\mathbf{x}_{c_i})+S_2(\mathbf{x}'_{c_i},\mathbf{x}_{c_i})\\ &= \frac{1}{N_c}\sum_{i=1}^{N_c}\big(2-S_1(\mathbf{x}'_{c_i},\mathbf{x}_{c_i})-S_2(\mathbf{x}'_{c_i},\mathbf{x}_{c_i})\big)\\ &= \frac{1}{N_c}\sum_{i=1}^{N_c}\tilde{g}^{x_{c_i}}(\mathbf{x}'_{c_i},\mathbf{x}_{c_i})^2.\end{aligned}$$

Since each term is convex within the range $\mathbf{x}'_{c_i}\in\mathcal{Q}_{\mathbf{x}_{c_i}}$, we can see that $\tilde{g}^x_{_M}(\mathbf{x}',\mathbf{x})^2$ is a convex function of $\mathbf{x}'$ within $\mathcal{Q}^M_{\mathbf{x}}$. Similarly, one can observe that for all positive values of h

$$\mathcal{Q}^M_{\mathbf{x}}\cap\Big\{\mathbf{z}\;\Big|\;\frac{1}{N_c}\sum_{i=1}^{N_c}\tilde{g}^{x_{c_i}}(\mathbf{z}_{c_i},\mathbf{x}_{c_i})^2\leqslant h\Big\}$$

and

$$\mathcal{Q}^M_{\mathbf{x}}\cap\Big\{\mathbf{z}\;\Big|\;\tilde{g}_{_M}(\mathbf{z},\mathbf{x})\leqslant\sqrt{h}\Big\},$$

are convex, which by definition implies the quasi-convexity of $\tilde{g}^x_{_M}(\mathbf{x}',\mathbf{x})$ restricted to $\mathbf{x}'\in\mathcal{Q}^M_x$.

For the sake of clarity and notational simplicity, we consider $\tilde{g}^x$ in our theoretical parts, i.e., Sections III, IV, and III-A and proceed with $\tilde{g}^x_{_M}$ in Sections V and VI, where we concentrate on near-optimal implementation of a minimization method. In other words, our theoretical results are valid for $\tilde{g}^x$, while our introduced method aims at optimizing $\tilde{g}^x_{_M}$.

III. Problem Statement

In image CF, the goal is to modify an image with minimal changes. Mathematically, an original image is replaced with another image in a group of images such that the replaced one maximally mimics the original one based on a chosen similarity metric.

Let $\mathcal{X}\subset\mathbb{R}^N$ denote the space of images in the vectorial form. A basic tool in every forensics system is the detector $\phi_x:\mathcal{X}\mapsto\{0,1\}$, that decides between two alternative hypotheses H_0 revealing $\mathbf{x}\in\mathcal{X}$ being "fake/modified" and H_1 meaning $\mathbf{x}\in\mathcal{X}$ is "original". Therefore, the space $\mathcal{X}$ is partitioned into two subsets R^x_0, R^x_1 as

$$R^x_k=\{\mathbf{x}\in\mathcal{X}:\phi_x(\mathbf{x})=k\},k=\{0,1\}. \quad (9)$$

In many scenarios, $\phi_x(\mathbf{x})$ has a simple characterization in terms of a transformed version of $\mathbf{x}$. This implies that $\phi_x(\mathbf{x})=\phi_y(f(\mathbf{x}))$, where $f:\mathcal{X}\mapsto\mathcal{Y}$ is a bijection and $\phi_y:\mathcal{Y}\mapsto\{0,1\}$ is an indicator function over $\mathcal{Y}\subset\mathbb{C}^N$. Among the widely-used examples of f one can name the linear transforms such as DFT, DCT and wavelets. The partitioning of $\mathcal{X}$ automatically induces a partitioning of $\mathcal{Y}$ into R^y_0 and R^y_1. In forensics, it is usually desirable to have one or a group of "fake/modified" images that resemble a given "original" image. To elaborate, we explain the main challenge in Problem 1.

Problem 1. *Given an original image $\mathbf{x}\in R^x_1$ and a similarity index $\Delta:\mathcal{X}\times\mathcal{X}\mapsto\mathbb{R}$, find*

$$\mathbf{x}^*=\underset{\mathbf{x}'\in R^x_0}{\operatorname{argmin}}\,\Delta(\mathbf{x}',\mathbf{x}). \quad (10)$$

Since R^x_0 and R^x_1 might not have a simple structure, finding the solution to Problem 1 can potentially be a combinatorial search. In Section IV, we try to narrow down the search domain for $\Delta\equiv\tilde{g}^x$, which might lead to substantial computational gain.

A. Histogram-based detector

Investigating the histogram of the data, particularly in the domain of discrete Fourier transform (DFT), has been one of the successful detection approaches [33], which is the center of our focus in this work. Taking the histogram into account, a reliable counter-forensics method shall leave the histogram of the image or its Fourier transform almost unchanged. Let $f(.)$ represent the full-frame DFT transform. According to the bijective nature of this $f(.)$, Problem 1 in terms of $\mathbf{x}$ can be translated in terms of $\mathbf{y}$ as (ignoring the histogram constraint)

$$\mathbf{y}^*=\underset{\mathbf{y}'\in R^y_0}{\operatorname{argmin}}\,\tilde{g}^x\big(f^{-1}(\mathbf{y}'),f^{-1}(\mathbf{y})\big), \quad (11)$$

where $\mathbf{y}=f(\mathbf{x})$. To account for the histogram, we indicate the set of histogram bin points by $\mathcal{B}=\{b_0,\ldots,b_{n_1}\}$, in which $b_0<b_1<\ldots<b_{n_1}$. This way, the histogram of $\mathbf{x}$ in terms of $\mathcal{B}$ shown by

$$H(\mathcal{B},\mathbf{x})\doteq[H(b_0,\mathbf{x}),\ldots,H(b_{n_1},\mathbf{x})]$$

is given by

$$H(b_i,\mathbf{x})\doteq\begin{cases}\frac{1}{N}\sum_{j=1}^N\mathbb{1}(x_j<b_0), & i=0,\\ \frac{1}{N}\sum_{j=1}^N\mathbb{1}(b_{i-1}\leqslant x_j<b_i), & 1\leq i<n_1,\\ \frac{1}{N}\sum_{j=1}^N\mathbb{1}(b_{n_1}\leqslant x_j), & i=n_1.\end{cases} \quad (12)$$

Now, the optimal histogram-based CF can be found via [5]:

$$H(\mathcal{B},\mathbf{y}^\sharp)=\underset{H(\mathcal{B},\mathbf{y}')\in\mathcal{R}^H_0}{\operatorname{argmin}}\,\tilde{g}^H\big(H(\mathcal{B},\mathbf{y}),H(\mathcal{B},\mathbf{y}')\big), \quad (13)$$

$$\mathbf{y}^*=\underset{\substack{\mathbf{y}'\\ H(\mathcal{B},\mathbf{y}')=H(\mathcal{B},\mathbf{y}^\sharp)}}{\operatorname{argmin}}\;\tilde{g}^y(\mathbf{y}',\mathbf{y}), \quad (14)$$

where $\tilde{g}^H$ measures the perceptual difference between histograms and $\tilde{g}^y$ quantifies the perceptual difference between two images in the DFT domain; specifically,

$$\tilde{g}^H\big(H(\mathcal{B},\mathbf{y}),H(\mathcal{B},\mathbf{y}')\big) \doteq \min_{\substack{\mathbf{y}'' \\ H(\mathcal{B},\mathbf{y}'')=H(\mathcal{B},\mathbf{y}')}} \tilde{g}^y(\mathbf{y}'',\mathbf{y}),$$
$$\tilde{g}^y(\mathbf{y}',\mathbf{y}) = \tilde{g}^x(f^{-1}(\mathbf{y}'),f^{-1}(\mathbf{y})). \quad (15)$$

Although our general approach in this paper is similar to [5] (as described above), rather than the MSE distance considered in [5], we base our $\tilde{g}^x$ and $\tilde{g}^x_M$ metrics on SSIM and MSSIM, respectively. Since SSIM and MSSIM do not have the component-wise additivity property [32], we can not simplify the numerical optimizations, as opposed to what has been done in [5].

IV. Locating the optimal solution

In this section, by taking advantage of quasi-convexity of $\tilde{g}^x_{_M}(.,\mathbf{x})$ stated in Lemma 1, we try to determine the minimizer of (10). Intuitively, we show that, the most similar element of R_0^x to a given $\mathbf{x} \in R_1^x$ lies at the border of the two regions. To have a precise statement, we first define border points.

Definition 1. *The detection boundary $\mathcal{C}$ is the set of all points $\mathbf{x} \in R_0^x \cup R_1^x$ that are arbitrarily close to both R_0^x and R_1^x. More precisely, $\mathbf{x}$ belongs to $\mathcal{C}$ if and only if for all $\epsilon > 0$, there exist $\mathbf{x}' \in R_0^x$ and $\mathbf{x}'' \in R_1^x$ such that*

$$\|\mathbf{x}-\mathbf{x}'\|, \|\mathbf{x}-\mathbf{x}''\| < \epsilon$$

To further proceed with finding the solution to Problem 1, we consider two scenarios based on the distance of $\mathbf{x}$ to the boundary $\mathcal{C}$:

(i) The point $\mathbf{x}$ is far from the boundary such that the quasi-convexity region $\mathcal{Q}_\mathbf{x}$ of $\tilde{g}^x(.,\mathbf{x})$ does not intersect $\mathcal{C}$; i.e., $\mathcal{Q}_\mathbf{x} \cap \mathcal{C} = \emptyset$.

(ii) The point $\mathbf{x}$ is so close to the boundary such that the quasi-convexity region of $\mathbf{x}$, $\mathcal{Q}_\mathbf{x}$, contains at least one boundary point, i.e., $\mathcal{Q}_\mathbf{x} \cap \mathcal{C} \neq \emptyset$.

The scenario (i) corresponds to the case where $\mathbf{x}$ and its immediate surroundings (at least those in $\mathcal{Q}_\mathbf{x}$) are located deep inside R_1^x. Consequently, the closest or most similar element of R_0^x to $\mathbf{x}$ is still quite far from it. A simple calculation shows that by evaluating the similarity of $\mathbf{x}$ with any $\mathbf{x}' \notin \mathcal{Q}_\mathbf{x}$, we achieve an SSIM value no better than 0.73, as the $\tilde{g}^x$ dissimilarity index exceeds $\frac{\sqrt{3}-1}{\sqrt{2}} \approx 0.52$ (shown in Appendix B) and consequently,

$$\text{SSIM} = S_1(\mathbf{x}',\mathbf{x}) \cdot S_2(\mathbf{x}',\mathbf{x}) \leq \left(\tfrac{S_1(\mathbf{x}',\mathbf{x})+S_2(\mathbf{x}',\mathbf{x})}{2}\right)^2$$
$$= \left(1 - \tfrac{(\tilde{g}^x(\mathbf{x}',\mathbf{x}))^2}{2}\right)^2 \leq 0.73, \quad (16)$$

which is considerably low and is very likely to be perceptually detectable. Therefore, scenario (i) presents a rather non-interesting setting for the purpose of forensics. As case (ii) is the only relevant case for forensics, we assume our setting falls in this category.

Theorem 1. *For $\Delta \equiv \tilde{g}^x$, if* (10) *has a minimizer inside the quasi-convexity region $\mathcal{Q}_\mathbf{x}$, then, it also has a minimizer in $\mathcal{C} \cap \mathcal{Q}_\mathbf{x}$.*

Proof. See Appendix C. ■

Theorem 1 helps in constraining the solution to Problem 1 in the sense that by knowing the boundary, we no longer need to inspect other points. However, there are rare cases in practice for which we know the boundary beforehand, or we can explicitly evaluate the boundary. Usually, the available tools can only check the values of $\phi_x(.)$ by examining the points one-by-one and predicting the boundary.

One possible approach to benefit from Theorem 1 is by combining the so-called optimal counter-forensics method (or in short, OCM) [5] with the blind Newton search [34]. For the latter to work, the initial point shall be in the intersection of the boundary and the convexity region around the image that contains $\mathcal{Q}_\mathbf{x}$. As identification of the boundary is itself a challenge, we employ OCM based on a simplified version of Problem 1 where the SSIM perceptual metric is replaced with the MSE. In line with the arguments above, we expect the output of OCM to be inside the quasi-convexity region $\mathcal{Q}_\mathbf{x}$ (otherwise, the MSE takes extremely poor values). Furthermore, it is proven in [5] that OCM always finds a point on the boundary $\mathcal{C}$. All in all, the result of OCM can be used as an initial point in the blind Newton search method.

While the combination of OCM and blind Newton search could potentially solve Problem 1, the overall computational cost is greatly affected by the structure of the boundary, particularly, when the regions are non-convex (i.e. R_0^x, R_1^x). In the next section, we define the common histogram-based detector and then, we propose a simple but sub-optimal solution to Problem 1.

V. Suboptimal Methods

As mentioned earlier, with the aid of Theorem 1, we can considerably restrict the search domain for the optimal solution and offer a preliminary point; however, the computational cost is still unfeasible in many practical cases. One possible remedy is to sacrifice optimality in return for computational cost reduction. What we propose is to find the optimal solution whenever the computational budget allows; for other scenarios, we introduce close-to-optimal solutions with low computational-complexity. We should emphasize that both the optimal and close-to-optimal methods are beneficial in special cases and could be tailored for specific problems.

To find the suboptimal answer of Problem 1 in the case of histogram-based detector, we propose two methods in the following subsections. In the first method, we exploit the concavity property of the low-order approximation to MSSIM (4) and propose a method by combining the OCM with the gradient descent algorithm; we call this method the gradient descent OCM (GDOCM).

For the second method, we make use of the fact that each permutation of $\mathbf{x}'$ can be represented by $U\mathbf{x}'$, where U is a permutation matrix (a binary-valued matrix with exactly a single 1 in each row and columns). Thus, instead of finding $\mathbf{x}^*$, we can look for U^* such that $\mathbf{x}^* = U^*\mathbf{x}'$. However, minimization problems over the set of permutation matrices are combinatorial in nature. Therefore, we relax this constraint and widen our search-space to the set of all unitary matrices

TABLE I. UMM algorithm with optional orthogonal projection

- Inputs
  1. $\mathbf{x}$, β, μ;
- Initialization
  1. $k = 0$;
  2. $W_0 = \mathrm{I}$;
  3. $\lambda = 1$;
- Compute the gradient of $\mathcal{T}(\cdot; \mu, \lambda)$:
  1. $\Gamma = \frac{\partial \mathcal{T}}{\partial W}(W_k)$
     $= \nabla_W \tilde{g}_M^x(W_k\mathbf{x}, \mathbf{x}) + \mu\big(\sum_{i,j=1}^N W(i,j) - N\big) - 2\lambda$;
- Choose the gradient descent method:
  1. If using the Riemannian gradient descent:
     a. Project the gradient direction onto the Reimannian space
        i. $G = \Gamma W_k^H - W_k \Gamma^H$;
     b. Determine the rotation Matrix:
        i. $P = \exp(-\beta G)$;
     c. Update the unitary matrix
        i. $W_{k+1} = W_k P$;
  2. If using ordinary gradient descent with orthogonal projection:
     a. Perform ordinary gradient descent
        i. $W_{k+1} = W_k - \beta\Gamma$;
     b. Project onto the manifold of orthogonal matrices
        i. Set $W_{k+1} = \text{Orthogonalize}(W_k)$;
- Update the augmented penalty parameter
  1. $\lambda = \lambda - \frac{\mu}{N}\Big(\sum_{i=1}^N \sum_{j=1}^N W_{k+1}(i,j) \ - N\Big)$;
- Increment the iteration counter
  1. $k = k + 1$;

with unit column and row sums. The row and column sum constraints are imposed to obtain unitary matrices closer to the set of permutation matrices. Denoting the set of unitary matrices with unit column and row sums by $\mathcal{W}$, we are searching for $W^* \in \mathcal{W}$ such that $\mathbf{x}^* = W^*\mathbf{x}'$; we call this second approach the unitary matrices method (UMM).

A. Gradient Descend OCM

In this subsection, we introduce GDOCM and then, compare its performance with the optimal exhaustive search in some toy examples (in particular, 4×3-sized images).

Let $\mathbf{x}^* = \mathcal{M}(\mathbf{x}', \mathbf{x})$ denote the permuted version of $\mathbf{x}'$ which is closest to $\mathbf{x}$ based on the Euclidean distance (MSE criterion) [35]. Therefore, if π and ϱ are the ordering permutations of $\mathbf{x}'$ and $\mathbf{x}^*$, respectively (meaning $x'_{\pi(1)} \leq \cdots \leq x'_{\pi(N)}$ and $x^*_{\varrho(1)} \leq \cdots \leq x^*_{\varrho(N)}$), then, we shall have $x^*_{\varrho(i)} = x'_{\pi(i)}$ for $i \in 1, \ldots, N$ [5]. Now that we can compute $\mathcal{M}(\mathbf{x}', \mathbf{x})$, we propose GDOCM ($\mathbf{x}^* = \mathcal{G}(\mathbf{x})$) as mentioned in Table II; in particular, this method relies on descent steps of $\tilde{g}_M^x(.,.)$ metric given by

$$\mathcal{A}(\mathbf{x}_1, \mathbf{x}, \mu) = \mathbf{x}_1 - \mu \cdot \nabla_{\mathbf{x}_1} \tilde{g}_M^x(\mathbf{x}_1, \mathbf{x}), \tag{17}$$

where μ is the step size. As explained in Table II, the method is initialized by the OCM solution, and the descent steps are applied until $\mathbf{x}_3$ experiences no update within 20 consecutive iterations (this number can be modified). For the DFT version of the GDOCM ($\mathbf{x}^* = \mathcal{G}_F(\mathbf{x})$), we need to replace the 3rd step in Table II with

$$\mathcal{F}\{\mathbf{x}_2\}_{\mathcal{L}} = \mathcal{M}\left(\mathcal{F}\{\mathbf{x}'\}_{\mathcal{L}}, \mathcal{F}\{\mathbf{x}_1\}_{\mathcal{L}}\right), \tag{18}$$

TABLE II. GDOCM algorithm

- Initialization
  1. $\mathbf{x}' =$ the OCM solution
     (will not be changed during algorithm);
  2. $\mathbf{x}_1 = \mathbf{x}_2 = \mathbf{x}_3 = \mathbf{x}'$, $c = 0$;
- While $\big(\tilde{g}_M^x(\mathbf{x}_2, \mathbf{x}) \neq 0$ and $c < 21\ \big)$
  1. Apply a descent step: $\mathbf{x}_1 = \mathcal{A}(\mathbf{x}_1, \mathbf{x}, \mu)$;
  2. Increase the counter: $c = c + 1$;
  3. Permuting $\mathbf{x}'$: $\mathbf{x}_2 = \mathcal{M}(\mathbf{x}', \mathbf{x}_1)$;
  4. If $\tilde{g}_M^x(\mathbf{x}_2, \mathbf{x}) \leq \tilde{g}_M^x(\mathbf{x}_3, \mathbf{x})$
     $\mathbf{x}_3 = \mathbf{x}_2$;
     $c = 0$;
- $\mathbf{x}^* = \mathbf{x}_3$;

where $\mathcal{L}$ indicates the set of specific frequencies considered by the detector [5].

Additionally, while this paper focuses on maximizing perceptual metrics like the mean structural similarity index (MSSIM), a GDOCM or UMM-like algorithm could also generate adversarial examples that preserve the histogram. By using adversarial loss $l(\mathbf{x}')$ instead of the perceptual metric $\tilde{g}(\mathbf{x}, \mathbf{x}')$, without requiring the original image $\mathbf{x}$ as a prior, this approach could fool specific forensics detectors while keeping modifications imperceptible. However, we leave this exploration for future work.

B. Unitary Matrices Method

In this method, we enlarge the feasible domain from permutation matrices to unitary matrices with unit column and row sums (the set $\mathcal{W}$),

$$\mathcal{W} = \Big\{W \,\Big|\, W^H W = W W^H = \mathrm{I}, \tag{19}$$

$$\sum_{i=1}^N W(i,j) = \sum_{j=1}^N W(i,j) = 1\Big\}, \tag{20}$$

and look for the optimal solution in the enlarged domain based on the descent algorithm:

$$W^* = \underset{W \in \mathcal{W}}{\operatorname{argmin}}\, \tilde{g}_M^x\big(W\,\mathbf{x}', \mathbf{x}\big), \tag{21}$$

$$\mathbf{x}^* = W^*\mathbf{x}'. \tag{22}$$

For solving the above optimization, we combine the augmented Lagrangian method [36] and the steepest descent algorithms for optimization under unitary matrix constraint [37]. To elaborate, we consider

$$\begin{aligned}
&\mathcal{T}(W; \mu, \lambda) = \tilde{g}_M^x\left(W\mathbf{x}', \mathbf{x}\right) + \\
&\mu\Big[\sum_{i=1}^N \Big(\sum_{j=1}^N W(i,j) \ - 1\Big)^2 + \sum_{j=1}^N \Big(\sum_{i=1}^N W(i,j) \ - 1\Big)^2\Big] \\
&- \lambda\Big[\sum_{i=1}^N \Big(\sum_{j=1}^N W(i,j) \ - 1\Big) + \sum_{j=1}^N \Big(\sum_{i=1}^N W(i,j) \ - 1\Big)\Big],
\end{aligned} \tag{23}$$

as our cost function, in which λ and μ are penalty parameters in the augmented Lagrangian method. Inspired by the method

in [37], we propose the UMM technique as explained in Table I. Similar to the GDOCM method, we first find the gradient of the augmented cost $\mathcal{T}(\cdot;\mu,\lambda)$ (called Γ); next, we project it onto the Reimannian space (G) and translate it into a rotation matrix P by scaling (using a given scalar β) and exponentiation. Moreover, instead of using Riemannian gradient descent to maintain orthogonality, one could employ ordinary gradient descent and subsequently project the result onto the manifold of orthogonal matrices. This projection can be done exactly by performing a singular value decomposition ($\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H \rightarrow \mathbf{W} = \mathbf{U}\mathbf{V}^H$) or through approximate iterative orthogonalization processes [38], [39] . Then, we update the solution and the penalty parameter λ.

Considering the $\mathcal{O}(n^3)$ computational complexity and $\mathcal{O}(n^2)$ memory requirements, where $n = w \times h$, GDOCM is preferred for larger images.

VI. Simulation Results

To investigate the efficacy of the proposed methods, we consider three simulation setups. In the first experiment, we generate random images with small sizes and compare all the methods including the exhaustive search. Next, we apply the methods on the logo of the Mozila Firefox browser as a representative of a real but small image. Finally, we consider the experimental setup in [5] consisting of real images.

A. Random images

In the first experiment, we generate random 3×4-pixel images[2] denoted in vectorial form by $\mathbf{x}$. We denote the double JPEG-compressed version of $\mathbf{x}$ by $\mathbf{x}'$ while fixing the quality factors of f_1, f_2 for the first and second JPEG-compression. Also, let $\mathcal{S}_{\mathbf{x}'}$ indicate the set of all permutations of $\mathbf{x}'$. We now have

$$R_k^x \doteq \{\mathbf{x} \in \mathbb{R}^{12} : \phi_x(\mathbf{x}, \mathbf{x}') = k\}, k = 0, 1, \tag{24}$$

$$\phi_x(\mathbf{x}, \mathbf{x}') = \begin{cases} 1; & \mathbf{x} \in \mathcal{S}_{\mathbf{x}'} \\ 0; & \mathbf{x} \notin \mathcal{S}_{\mathbf{x}'} \end{cases}. \tag{25}$$

The MSSIM metric of the results obtained by the MSE method (OCM), GDOCM and exhaustive search (ES) which are formulated as

$$\mathbf{x}_{\text{OCM}} = \underset{\mathbf{x}'' \in \mathcal{S}_{\mathbf{x}'}}{\operatorname{argmin}} \operatorname{MSE}(\mathbf{x}'', \mathbf{x}), \tag{26}$$

$$\mathbf{x}_{\text{GDOCM}} = \underset{\mathbf{x}'' \in \mathcal{S}_{\mathbf{x}'}}{\operatorname{argmin}} \tilde{g}_M^x(\mathbf{x}'', \mathbf{x}), \tag{27}$$

$$\mathbf{x}_{\text{ES}} = \underset{\mathbf{x}'' \in \mathcal{S}_{\mathbf{x}'}}{\operatorname{argmax}} \operatorname{MSSIM}(\mathbf{x}'', \mathbf{x}), \tag{28}$$

are reported in Table III for different f_1 and f_2 quality factors in the first and second compressions, respectively. Obviously, the exhaustive search method in (28) yields the highest quality. It is worth mentioning that for this method, we checked for the highest MSSIM metric among all possible permutation of $\mathbf{x}'$. Due to the computational cost (checking $N!$ permutations, where N is the overall number of pixels), this approach is impractical in moderate to high dimensions. With this point of view, (26) and (27) could be considered as computationally feasible approximations of (28). The results reported in Table III clearly distinguish GDOCM as the superior approximate. Besides, the results of GDOCM closely follow the optimal results by the ES method.

[2]The high computational cost of the optimal exhaustive search method is the rationale behind considering such toy images.

TABLE III. The average MSSIM metric for the OCM (MSE), GDOCM and ES methods over 50 random 3×4 images.

	f_2	f_1	**OCM**	**GDOCM**	**ES**
Scenario 1	70	10	0.8715	**0.9133**	0.9229
Scenario 2	70	20	0.8883	**0.9215**	0.9331
Scenario 3	70	30	0.9023	**0.9297**	0.9377
Scenario 4	70	40	0.9105	**0.9368**	0.9414
Scenario 5	70	50	0.9180	**0.9412**	0.9433
Scenario 6	70	60	0.9219	**0.9531**	0.9596

TABLE IV. Average wall clock time, memory usage, and complexity over 50 random 3×4 images. experiments done using an x86_64 CPU, and an NVIDIA A100 80GB GPU which was heavily utilised for ES. Here R denotes the maximum number of iterations utilized for each algorithm.

	OCM	**GDOCM**	**UMM**	**ES**
Runtime	0.4 ms	11.57 s	23.10 s	910.35 s
Memory usage	0.12 KiB	302.40 KiB	319.70 KiB	284.31 KiB
Complexity	$\mathcal{O}(n \log n)$	$\mathcal{O}(R \cdot n \log n)$	$\mathcal{O}(R \cdot n^3)$	$\mathcal{O}(n!)$
Memory	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n)$

We acknowledge that our initial experiments focused on small synthetic datasets, such as 3×4 pixel images, due to the computational infeasibility of the exhaustive search (ES) algorithm, which scales with $\mathcal{O}(n!)$. This approach was necessary for theoretical validation but has limited practical applicability. To address this, we have expanded our experiments to include larger, more complex natural images for evaluating double JPEG compression distortion, excluding the ES algorithm. This shift allows us to demonstrate the practical utility of our methods on more realistic datasets, providing robust evidence of their effectiveness in real-world scenarios.

B. Mozila Firefox logo

As a more realistic image, we experiment on the logo of the "Mozilla Firefox" browser with size ranging from16×16 to 64×64; we refer to the vectorial form of the non-distorted and original logo by $\mathbf{x}$. We further add white Gaussian noise to form the fake image $\mathbf{x}' = \mathbf{x} + \mathbf{s}$, in which $\mathbf{s}$ stands for the added noise. According to (24), our goal is to find the most similar image to the original logo among all permutations of the distorted image (i.e., $\mathbf{x}'$). In Figure 1, we plot the resulting MSSIM metrics for the OCM method, as well as the proposed GDOCM and UMM methods, under different noise levels, when the logo of size 16×16 is considered. In Figures 2 and 3, we plot similar curves when the logo size is changed to 32×32 and 64×64, respectively. All these figures confirm that GDOCM and UMM have similar performances considerably better than the OCM method. Another interesting observation is that although GDOCM is computationally less expensive

TABLE V. The MSSIM and wall clock time for the OCM, GDOCM, and UMM methods in the Block-DCT domain, evaluated on natural images resized to various dimensions under Double JPEG Compression.

	MSSIM			Wall Clock Time (s)		
Image size	**OCM**	**GDOCM**	**UMM**	**OCM**	**GDOCM**	**UMM**
12×12	0.9636	0.9709	0.9863	0.0015	0.5385	54.64
32×32	0.9872	0.9919	0.9951	0.0046	1.2799	104.32
64×64	0.9651	0.9891	—	0.0094	1.4093	—
128×128	0.9258	0.9785	—	0.0254	2.1092	—
256×256	0.8787	0.9645	—	0.0356	2.8716	—
512×512	0.8239	0.9542	—	0.0900	5.7799	—
1024×1024	0.9266	0.9793	—	0.2601	5.0552	—

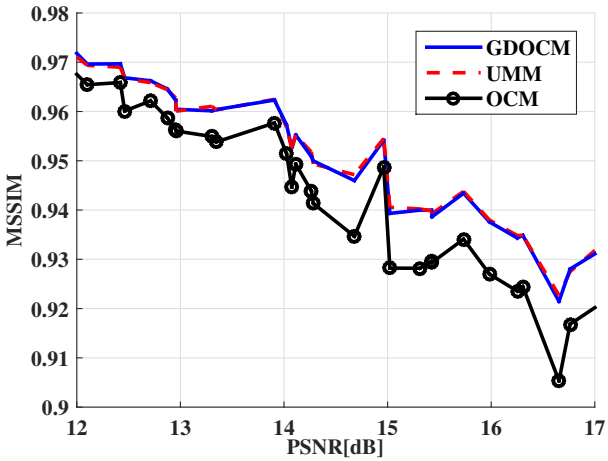

Fig. 1. MSSIM comparison of the methods applied to the 16×16 Mozila Firefox logo: GDOCM: $\mu = 2 \times 10^{-7}$, UMM: $\beta = 2 \times 10^{-7}$, $\mu = 100$, and applying the Riemannian gradient descent.

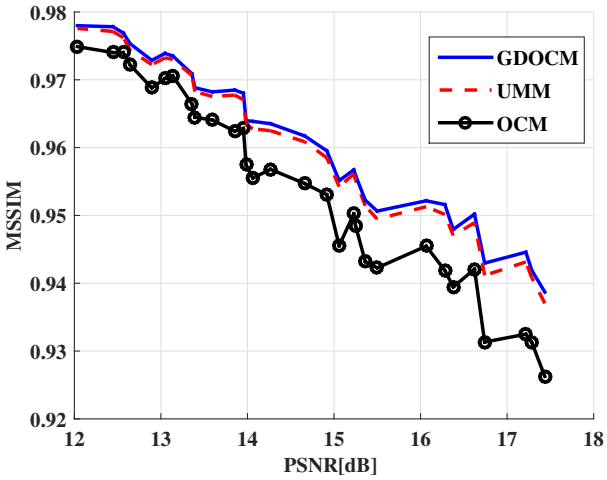

Fig. 2. MSSIM comparison of the methods applied to the 32×32 Mozila Firefox logo: GDOCM: $\mu = 3.5 \times 10^{-7}$, UMM: $\beta = 2 \times 10^{-7}$, $\mu = 100$, and applying the Riemannian gradient descent.

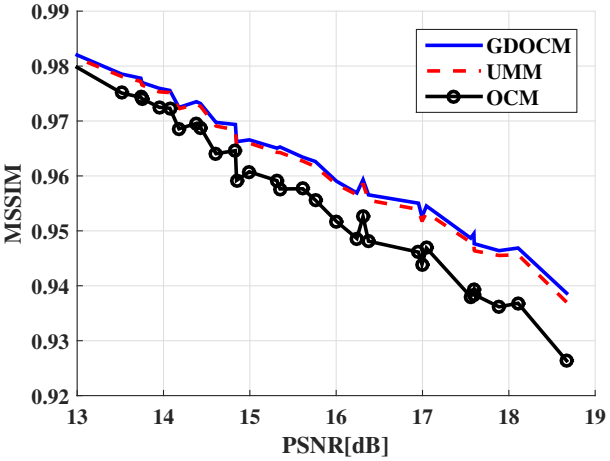

Fig. 3. MSSIM comparison of the methods applied to the 64×64 Mozila Firefox logo: GDOCM: $\mu = 5 \times 10^{-7}$, UMM: $\beta = 2 \times 10^{-7}$, $\mu = 100$, and applying the Riemannian gradient descent.

than UMM (the latter requires N^2-sized matrices, while the former fully operates in dimension N), it exhibits a marginally better performance, particularly, with larger images. For better illustration of the results, we have sketched the outcomes in Figure 4 for the 64×64 logo and three different noise levels.

C. Test on real images

Finally, we experiment on real images provided in the UCIDv2 image database [40] and compare OCM with the proposed GDOCM. We provide the same setup as in the experimental section in [5]; in particular, we include the double JPEG-compression detector proposed in [33] in contrast to (24). As illustrated in Table VI, GDOCM outperforms OCM in terms of MSSIM, while OCM outperforms GDOCM in terms of PSNR (or MSE).

VII. Conclusion

In this work, based on an introduced quasi-convex approximation of SSIM, we considered the counter-forensics problem. In particular, we showed that the optimal solution based on this approximation lies on a specific region that we call boundary. This results, considerably restricts the search area. To practically find a solution, we proposed two computational tractable but sub-optimal methods. Simulations results show that both methods outperform the conventional OCM method.

Appendix A
Region of Quasi-Convexity

According to Section II-C of [31], we have that

$$\begin{aligned} \tilde{g}^x(\mathbf{x}', \mathbf{x}) &= \|\mathbf{d}(\mathbf{x}, \mathbf{x}')\|_2 \\ &= \sqrt{d_1(\overline{\mathbf{x}}, \overline{\mathbf{x}}')^2 + d_2(\mathbf{x} - \overline{\mathbf{x}}, \mathbf{x}' - \overline{\mathbf{x}}')^2} \end{aligned} \tag{29}$$

t

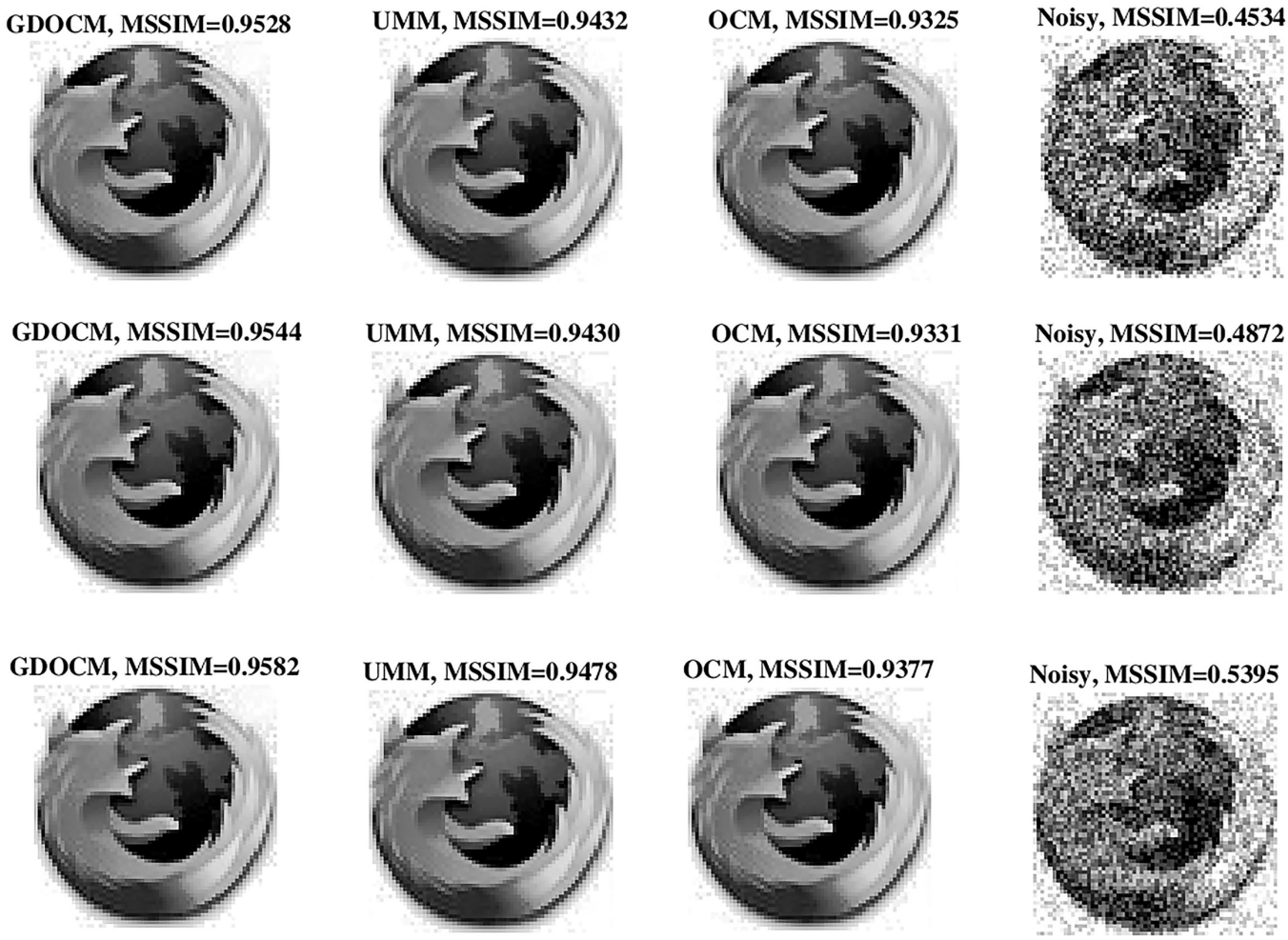

Fig. 4. The reconstructed 64×64 Mozila Firefox logo based on the noisy images by the OCM, GDOCM and UMM methods at different noise levels; the first, second and third rows correspond to noisy images with PSNR=14.7067dB, PSNR=15.5946dB, PSNR=16.6654dB, respectively.

Fig. 5. Reconstruction of the 512×512 Stanford logo in the Block-DCT domain, with MSSIM values indicating structural similarity: (Attacked) 0.7539, (OCM) 0.9008, (GDOCM) 0.9618.

TABLE VI. Comparing the results of OCM and GDOCM in terms of MSSIM and PSNR (averaged over a selection of UCIDv2 images). Different rows have different added noise levels.

	PSNR (OCM)	PSNR (GDOCM)	MSSIM (OCM)	MSSIM (GDOCM)
1	30.9	30.5	0.913	0.928
2	32.9	32.6	0.921	0.932
3	33.8	33.3	0.939	0.946
4	35.7	35.1	0.952	0.959
5	38.3	37.8	0.963	0.966
6	40.1	39.5	0.979	0.988

where d_1 and d_2 are defined below based on $\boldsymbol{\delta}_{\mathbf{x}} = \mathbf{x} - \overline{\mathbf{x}}$ and $\boldsymbol{\delta}_{\mathbf{x}'} = \mathbf{x}' - \overline{\mathbf{x}}'$:

$$d_1(\overline{\mathbf{x}}, \overline{\mathbf{x}}') = \sqrt{\frac{|\overline{\mathbf{x}} - \overline{\mathbf{x}}'|^2}{\overline{\mathbf{x}}^2 + \overline{\mathbf{x}}'^2 + c_1}} = \text{NRMSE}(\overline{\mathbf{x}}, \overline{\mathbf{x}}', c_1),$$

$$d_2(\boldsymbol{\delta}_{\mathbf{x}}, \boldsymbol{\delta}_{\mathbf{x}'}) = \sqrt{\frac{\|\boldsymbol{\delta}_{\mathbf{x}} - \boldsymbol{\delta}_{\mathbf{x}'}\|^2}{\|\boldsymbol{\delta}_{\mathbf{x}}\|^2 + \|\boldsymbol{\delta}_{\mathbf{x}'}\|^2 + (N-1)c_2}} = \text{NRMSE}(\boldsymbol{\delta}_{\mathbf{x}}, \boldsymbol{\delta}_{\mathbf{x}'}, c_2).$$

As discussed in [31], one can see that $\text{NRMSE}(\mathbf{x}, \mathbf{y}, 0)$ is convex with respect to $\mathbf{x}$ within the hyper-sphere

$$R_{\mathbf{y}} = \Big\{ \mathbf{x} \in \mathbb{R}^N_+ \;\Big|\; \|\mathbf{x} - \mathbf{y}\| \leq (\sqrt{3} - 1)\|\mathbf{y}\| \Big\}.$$

Since both $\overline{\mathbf{x}}$ and $\mathbf{x} - \overline{\mathbf{x}}$ are linearly related to $\mathbf{x}$, we know that $d_1(\overline{\mathbf{x}}, \overline{\mathbf{y}})$ and $d_2(\boldsymbol{\delta}_{\mathbf{x}}, \boldsymbol{\delta}_{\mathbf{y}})$ are simultaneously convex with respect $\mathbf{x}$ restricted to the curved hyper-cylinder $\mathcal{Q}_{\mathbf{y}}$

$$\mathcal{Q}_{\mathbf{y}} = \big\{ \mathbf{x} \,\big|\, \overline{\mathbf{x}} \in \mathcal{Q}_1 \,,\, \underbrace{\mathbf{x} - \overline{\mathbf{x}}}_{\boldsymbol{\delta}_{\mathbf{x}}} \in \mathcal{Q}_2 \big\}, \tag{30}$$

where

$$\begin{aligned} &\mathcal{Q}_1 = \big[0, \sqrt{3}\overline{\mathbf{y}}\big], \\ &\mathcal{Q}_2 = \Big\{ \boldsymbol{\delta}_{\mathbf{x}} \in \mathbb{R}^{N-1}_+ \;\Big|\; \|\boldsymbol{\delta}_{\mathbf{x}} - \boldsymbol{\delta}_{\mathbf{y}}\| \leq (\sqrt{3} - 1)\|\boldsymbol{\delta}_{\mathbf{y}}\| \Big\}, \\ &\boldsymbol{\delta}_{\mathbf{y}} = \mathbf{y} - \overline{\mathbf{y}}. \end{aligned} \tag{31}$$

As observed in [31], the actual convexity region of $\text{NRMSE}(\mathbf{x}, \mathbf{y}, 0)$ forms a teardrop shape that extends beyond $\mathcal{Q}_2$. Consequently, the simultaneous convexity region of $d_1(\overline{\mathbf{x}}, \overline{\mathbf{y}})$ and $d_2(\boldsymbol{\delta}_{\mathbf{x}}, \boldsymbol{\delta}_{\mathbf{y}})$ (with respect $\mathbf{x}$) exceeds the boundaries of $\mathcal{Q}_{\mathbf{y}}$. Because of the fact that the ℓ_2-norm operator $\|\cdot\|_2$ is both convex and non-decreasing with respect to each element, we can conclude that its composition with $d_1(\overline{\mathbf{x}}, \overline{\mathbf{y}})$ and $d_2(\boldsymbol{\delta}_{\mathbf{x}}, \boldsymbol{\delta}_{\mathbf{y}})$, i.e.,

$$\sqrt{|d_1(\overline{\mathbf{x}}, \overline{\mathbf{y}})|^2 + \|d_2(\boldsymbol{\delta}_{\mathbf{x}}, \boldsymbol{\delta}_{\mathbf{y}})\|^2} \tag{32}$$

is also convex, given $\mathbf{x} \in \mathcal{Q}_{\mathbf{y}}$. This conforms that $\tilde{g}^x(\mathbf{x}, \mathbf{x}')$ is convex on at-least $\mathcal{Q}_{\mathbf{x}}$:

$$\mathcal{Q}_{\mathbf{x}} = \Big\{ \mathbf{x}' \in \mathbb{R}^N \;\Big|\; \overline{\mathbf{x}}' \in [0, \sqrt{3}\overline{\mathbf{x}}], \; \tfrac{\|(\mathbf{x} - \overline{\mathbf{x}}) - (\mathbf{x}' - \overline{\mathbf{x}}')\|}{\|(\mathbf{x} - \overline{\mathbf{x}})\|} \leq \sqrt{3} - 1 \Big\}. \tag{33}$$

It is worth noting that $\tilde{g}^x(\mathbf{x}', \mathbf{x})^2 = 2 - S_1(\mathbf{x}', \mathbf{x}) - S_2(\mathbf{x}', \mathbf{x})$ and the vector (S_1, S_2) also exhibit convexity within this region, as demonstrated in the proof and corroborated by the findings in [31]. In our proof, we set $c_1 = 0$ and $c_2 = 0$; it is, however, common in practical applications to use small but nonzero values to avoid numerical instability.

B Upper Bound For The SSIM of the Points Outside the Quasi-Convexity Region

Here, we establish a lower bound on $\tilde{g}^x(\mathbf{x}', \mathbf{x})$ knowing that $\mathbf{x}' \notin \mathcal{Q}_{\mathbf{x}}$, which essentially means $\frac{\|\boldsymbol{\delta}' - \boldsymbol{\delta}\|}{\|\boldsymbol{\delta}\|} \geq \sqrt{3} - 1$ (where $\boldsymbol{\delta} = \mathbf{x} - \overline{\mathbf{x}}$ and $\boldsymbol{\delta}' = \mathbf{x}' - \overline{\mathbf{x}}'$) and that $\overline{\mathbf{x}}' \notin \big[0, \sqrt{3}\overline{\mathbf{x}}\big]$. Because of this constraint, we rewrite (29) as

$$\begin{aligned} \tilde{g}^x(\mathbf{x}', \mathbf{x})^2 &= \tfrac{|\overline{\mathbf{x}} - \overline{\mathbf{x}}'|^2}{\overline{\mathbf{x}}^2 + \overline{\mathbf{x}}'^2 + c_1} + \tfrac{\|\boldsymbol{\delta} - \boldsymbol{\delta}'\|^2}{\|\boldsymbol{\delta}\|^2 + \|\boldsymbol{\delta}'\|^2 + (N-1)c_2} \\ &= \frac{\frac{|\overline{\mathbf{x}} - \overline{\mathbf{x}}'|^2}{\overline{\mathbf{x}}^2}}{1 + \frac{\overline{\mathbf{x}}'^2}{\overline{\mathbf{x}}^2} + \frac{c_1}{\overline{\mathbf{x}}^2}} + \frac{\frac{\|\boldsymbol{\delta}' - \boldsymbol{\delta}\|^2}{\|\boldsymbol{\delta}\|^2}}{1 + \frac{\|\boldsymbol{\delta}'\|^2}{\|\boldsymbol{\delta}\|^2} + \frac{(N-1)c_2}{\|\mathbf{x}\|^2}}. \end{aligned} \tag{34}$$

Now, we can write

$$\tfrac{\overline{\mathbf{x}}'^2}{\overline{\mathbf{x}}^2} = \tfrac{[(\overline{\mathbf{x}}' - \overline{\mathbf{x}}) + \overline{\mathbf{x}}]^2}{\overline{\mathbf{x}}^2} = 1 + \tfrac{|\overline{\mathbf{x}} - \overline{\mathbf{x}}'|^2}{\overline{\mathbf{x}}^2} \pm 2\tfrac{|\overline{\mathbf{x}}' - \overline{\mathbf{x}}|}{|\overline{\mathbf{x}}|}, \tag{35}$$

and

$$\tfrac{\|\boldsymbol{\delta}'\|^2}{\|\boldsymbol{\delta}\|^2} = \tfrac{\|[(\boldsymbol{\delta}' - \boldsymbol{\delta}) + \boldsymbol{\delta}]\|^2}{\|\boldsymbol{\delta}\|^2} = 1 + \tfrac{\|\boldsymbol{\delta}' - \boldsymbol{\delta}\|^2}{\|\boldsymbol{\delta}\|^2} + 2\tfrac{\|\boldsymbol{\delta}' - \boldsymbol{\delta}\|}{\|\boldsymbol{\delta}\|}\cos(\theta), \tag{36}$$

where θ denotes the angle between $\boldsymbol{\delta}' - \boldsymbol{\delta}$ and $\boldsymbol{\delta}$. If we further define

$$\alpha = \frac{\|\boldsymbol{\delta}' - \boldsymbol{\delta}\|}{\|\boldsymbol{\delta}\|} > 0 \quad , \quad \beta = \frac{|\overline{\mathbf{x}}' - \overline{\mathbf{x}}|}{|\overline{\mathbf{x}}|} > 0, \tag{37}$$

we can reformulate (34) as

$$\tilde{g}^x(\mathbf{x}, \mathbf{x}') = \tfrac{\beta^2}{2 \pm 2\beta + \beta^2 + \frac{c_1}{\overline{\mathbf{x}}^2}} + \tfrac{\alpha^2}{2 + 2\alpha\cos(\theta) + \alpha^2 + \frac{(N-1)c_2}{\|\mathbf{x}\|^2}}. \tag{38}$$

If $c_1 = c_2 = 0$, or we can assume $\overline{\mathbf{x}}^2 \gg c_1, \|\mathbf{x}\|^2 \gg (N-1)c_2$, then, we can fairly approximate $\tilde{g}^x(\mathbf{x}, \mathbf{x}')$ via

$$\begin{aligned} \tilde{g}^x(\mathbf{x}, \mathbf{x}')^2 &\approx \frac{\alpha^2}{2 + 2\alpha\cos(\theta) + \alpha^2} + \frac{\beta^2}{2 \pm 2\beta + \beta^2} \\ &\geq \frac{1}{f(\beta)} + \frac{1}{f(\alpha)}, \end{aligned} \tag{39}$$

where $f(x) = \frac{x^2 + 2x + 2}{x^2} = 1 + \frac{2}{x} + \frac{2}{x^2}$ is a strictly decreasing function at $x > 0$. Moreover, we recall that $\mathbf{x}' \notin \mathcal{Q}_{\mathbf{x}}$ implies $\alpha, \beta \geq \sqrt{3} - 1$. Thus,

$$\begin{aligned} &\tilde{g}^x(\mathbf{x}', \mathbf{x})^2 \geq \frac{2}{f(\sqrt{3} - 1)} = (2 - \sqrt{3})^2 = \big(\tfrac{\sqrt{3} - 1}{\sqrt{2}}\big)^2 \\ &\Rightarrow \quad \tilde{g}^x(\mathbf{x}', \mathbf{x}) \geq \frac{\sqrt{3} - 1}{\sqrt{2}} \approx 0.52. \end{aligned} \tag{40}$$

C Proof of Theorem 1

To simplify the notations and the proof, we first state a definition and a lemma.

Besides the point of interest $\mathbf{x}$, let $\mathbf{x}'$ be any point inside the quasi-convexity region $\mathcal{Q}_{\mathbf{x}}$. We denote the line connecting $\mathbf{x}$ to $\mathbf{x}'$ by $L_{\mathbf{x},\mathbf{x}'}$:

$$L_{\mathbf{x},\mathbf{x}'} \doteq \left\{(1-\lambda)\mathbf{x} + \lambda\mathbf{x}' \;\middle|\; 0 \leqslant \lambda \leqslant 1\right\}. \tag{41}$$

As $\mathcal{Q}_{\mathbf{x}}$ is a curved hyper-cylinder (see (33) for the definition), it is a convex set. Thus, with $\mathbf{x}, \mathbf{x}' \in \mathcal{Q}_{\mathbf{x}}$ we are guaranteed that $L_{\mathbf{x},\mathbf{x}'} \subseteq \mathcal{Q}_{\mathbf{x}}$. We also define $\mathbf{z}_d^{L_{\mathbf{x},\mathbf{x}'}}$ as a point on $L_{\mathbf{x},\mathbf{x}'}$ with the normalized distance $d \in [0,1]$ from $\mathbf{x}'$:

$$\mathbf{z}_d^{L_{\mathbf{x},\mathbf{x}'}} = d\frac{\mathbf{x}-\mathbf{x}'}{\|\mathbf{x}-\mathbf{x}'\|} + \mathbf{x}'. \tag{42}$$

Lemma 3. *For all $\mathbf{z} \in L_{\mathbf{x},\mathbf{x}'}$ we have that,*

$$\tilde{g}^x(\mathbf{z},\mathbf{x}) \leqslant \tilde{g}^x(\mathbf{x}',\mathbf{x}). \tag{43}$$

Proof of Lemma 3. Contrary to the claim, let $\tilde{\mathbf{z}} \in L_{\mathbf{x},\mathbf{x}'}$ be such that

$$\tilde{g}^x(\tilde{\mathbf{z}},\mathbf{x}) > \tilde{g}^x(\mathbf{x}',\mathbf{x}).$$

Set $h = \frac{\tilde{g}^x(\tilde{\mathbf{z}},\mathbf{x})+\tilde{g}^x(\mathbf{x}',\mathbf{x})}{2}$. By invoking the continuity of $\tilde{g}^x(\cdot,\mathbf{x})$, we know that $h \in Range\big(\tilde{g}^x(\cdot,\mathbf{x})\big)$. Further, let

$$\mathcal{Q}_{\mathbf{x}}^h \doteq \left\{\mathbf{z} \in \mathcal{Q}_{\mathbf{x}} \;\middle|\; \tilde{g}^x(\mathbf{z},\mathbf{x}) \leqslant h\right\}. \tag{44}$$

Obviously, $\mathbf{x}, \mathbf{x}' \in \mathcal{Q}_{\mathbf{x}}$ and $\tilde{\mathbf{z}} \notin \mathcal{Q}_{\mathbf{x}}^h$. Thus, $\mathcal{Q}_{\mathbf{x}}^h$ includes two points such that their connecting line does not completely lie in $L_{\mathbf{x},\mathbf{x}'}$. This contradicts the quasi-convexity of $\tilde{g}^x(\cdot,\mathbf{x})$ inside $\mathcal{Q}_{\mathbf{x}}^h$ and consequently $\mathcal{Q}_{\mathbf{x}}$. ■

Now that we covered the required background, we get back to the proof of Theorem 1. Below, we prove the equivalent statement that, for each $\mathbf{x}' \in R_0^x \bigcap \mathcal{Q}_{\mathbf{x}}$ there exists $\mathbf{z} \in \mathcal{C} \bigcap \mathcal{Q}_{\mathbf{x}}$ such that:

$$\tilde{g}^x(\mathbf{z},\mathbf{x}) \leqslant \tilde{g}^x(\mathbf{x}',\mathbf{x}). \tag{45}$$

Hence, at least one of the minimizers of $\tilde{g}^x(\cdot,\mathbf{x})$ shall be on the boundary $\mathcal{C}$. Note that the minimum value of $\tilde{g}^x(\mathbf{x}',\mathbf{x})$ for $\mathbf{x}' \in \mathcal{C} \bigcap \mathcal{Q}_{\mathbf{x}}$ cannot be strictly less than the minimum value for $\mathbf{x}' \in R_0^x \bigcap \mathcal{Q}_{\mathbf{x}}$, because $\tilde{g}^x(\cdot,\mathbf{x})$ is continuous and any point in $\mathcal{C}$ is the limit of a sequence in R_0^x.

Proof of Theorem 1. Let us fix $\mathbf{x}' \in R_0^x \bigcap \mathcal{Q}_{\mathbf{x}}$. If we manage to show that the line $L_{\mathbf{x},\mathbf{x}'}$ intersects the boundary $\mathcal{C}$, then, the proof directly follows by applying Lemma 3. Here, we explicitly introduce one of the points in this intersection (note that $L_{\mathbf{x},\mathbf{x}'}$ and $\mathcal{C}$ can intersect in more than one point, particularly, when the R_0^x and R_1^x are non-convex). To this end, we consider the point on $L_{\mathbf{x},\mathbf{x}'}$ furthest from $\mathbf{x}'$ such that all the line segment connecting $\mathbf{x}'$ to this point fall inside R_0^x. More precisely, if

$$r \doteq \sup\left\{\frac{\|\mathbf{z}-\mathbf{x}\|}{\|\mathbf{x}'-\mathbf{x}\|} \;\middle|\; \mathbf{z} \in L_{\mathbf{x},\mathbf{x}'}, \underset{\forall 0\leqslant\lambda\leqslant 1}{(1-\lambda)\mathbf{x}+\lambda\mathbf{z} \in R_0^x}\right\}, \tag{46}$$

then, we show that $\mathbf{z}_r^{L_{\mathbf{x},\mathbf{x}'}}$ is on the boundary $\mathcal{C}$ and could be used as $\mathbf{z}$ in (45). For this reason, we separately treat the following three cases:

a) $r = 0$ or alternatively, $\mathbf{z}_r^{L_{\mathbf{x},\mathbf{x}'}} = \mathbf{x}$. The definition of r in (46) demonstrates that any ϵ-neighbourhood of $\mathbf{z}_r^{L_{\mathbf{x},\mathbf{x}'}}$ in $L_{\mathbf{x},\mathbf{x}'}$ contains a point in R_1^x. As $\mathbf{z}_r^{L_{\mathbf{x},\mathbf{x}'}} = \mathbf{x} \in R_0^x$, it is arbitrarily close to both R_0^x, R_1^x; i.e., $\mathbf{z}_r^{L_{\mathbf{x},\mathbf{x}'}} = \mathbf{x}$ is a boundary point.

b) $r = 1$ or alternatively, $\mathbf{z}_r^{L_{\mathbf{x},\mathbf{x}'}} = \mathbf{x}'$. Again, the definition of r in (46) implies that for all $\epsilon > 0$, the point $\mathbf{z}_{r-\epsilon}^{L_{\mathbf{x},\mathbf{x}'}}$ is only ϵ away (normalized distance) from $\mathbf{x}'$ and belongs to R_0^x. Since $\mathbf{x}' \in R_1^x$, we conclude that $\mathbf{z}_{r-\epsilon}^{L_{\mathbf{x},\mathbf{x}'}} = \mathbf{x}'$ is a boundary point.

c) $0 < r < 1$. Let ϵ be a small positive real, namely, $0 < \epsilon < \min(r, 1-r)$. Based on the definition of r, it is evident that $\mathbf{z}_{r-\frac{\epsilon}{2}}^{L_{\mathbf{x},\mathbf{x}'}} \in R_0^x$. Furthermore, the set

$$\left\{(1-\lambda)\,\mathbf{z}_{r-\frac{\epsilon}{2}}^{L_{\mathbf{x},\mathbf{x}'}} + \lambda\,\mathbf{z}_{r+\frac{\epsilon}{2}}^{L_{\mathbf{x},\mathbf{x}'}} \;\middle|\; 0 \leqslant \lambda \leqslant 1\right\}$$

shall intersect R_1^x; otherwise, r can be replaced with at least $r + \frac{\epsilon}{2}$ in the definition of (46). This completes the proof, as for any small-enough ϵ, the $\frac{\epsilon}{2}$ neighborhood of $\mathbf{z}_r^{L_{\mathbf{x},\mathbf{x}'}}$ intersects with both regions R_0^x and R_1^x.

■

References

[1] E. Nowroozi, A. Dehghantanha, R. M. Parizi, and K.-K. R. Choo, "A survey of machine learning techniques in adversarial image forensics," *Computers & Security*, vol. 100, p. 102092, 2021.

[2] S. Ferreira, M. Antunes, and M. E. Correia, "Exposing manipulated photos and videos in digital forensics analysis," *Journal of Imaging*, vol. 7, p. 102, 06 2021.

[3] M. Chawki, "Navigating legal challenges of deepfakes in the american context: a call to action," *Cogent Engineering*, vol. 11, no. 1, p. 2320971, 2024. [Online]. Available: https://doi.org/10.1080/23311916.2024.2320971

[4] M.-P. Sandoval, M. Vau, J. Solaas, and L. Rodrigues, "Threat of deepfakes to the criminal justice system: a systematic review," *Crime Science*, vol. 13, 11 2024.

[5] P. Comesana-Alfaro and F. Pérez-González, "Optimal counterforensics for histogram-based forensics," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process*, 2013, pp. 3048–3052.

[6] P. Korus, "Digital image integrity–a survey of protection and verification techniques," *Digital Signal Processing*, vol. 71, pp. 1–26, 2017.

[7] M. A. Qureshi and E.-S. M. El-Alfy, "Bibliography of digital image anti-forensics and anti-anti-forensics techniques," *IET Image Processing*, vol. 13, no. 11, pp. 1811–1823, 2019.

[8] P. Yang, D. Baracchi, R. Ni, Y. Zhao, F. Argenti, and A. Piva, "A survey of deep learning-based source image forensics," *Journal of Imaging*, vol. 6, no. 3, p. 9, 2020.

[9] M. C. Stamm, S. K. Tjoa, W. S. Lin, and K. R. Liu, "Anti-forensics of jpeg compression," in *2010 IEEE International conference on acoustics, speech and signal processing*. IEEE, 2010, pp. 1694–1697.

[10] G. Valenzise, M. Tagliasacchi, and S. Tubaro, "The cost of jpeg compression anti-forensics," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 1884–1887.

[11] W. Fan, K. Wang, F. Cayre, and Z. Xiong, "A variational approach to jpeg anti-forensics," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 3058–3062.

[12] S. Lai and R. Böhme, "Countering counter-forensics: The case of jpeg compression," in *International Workshop on Information Hiding*. Springer, 2011, pp. 285–298.

[13] W. Luo, Z. Qu, J. Huang, and G. Qiu, "A novel method for detecting cropped and recompressed image block," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, vol. 2. IEEE, 2007, pp. II–217.

[14] F. Huang, J. Huang, and Y. Q. Shi, "Detecting double jpeg compression with the same quantization matrix," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 848–856, 2010.

[15] T. Bianchi and A. Piva, "Detection of nonaligned double jpeg compression based on integer periodicity maps," *IEEE transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 842–848, 2011.

[16] M. Barni, Z. Chen, and B. Tondi, "Adversary-aware, data-driven detection of double jpeg compression: How to make counter-forensics harder," in *2016 IEEE international workshop on information forensics and security (WIFS)*. IEEE, 2016, pp. 1–6.

[17] G. Cao, Y. Zhao, R. Ni, and X. Li, "Contrast enhancement-based forensics in digital images," *IEEE transactions on information forensics and security*, vol. 9, no. 3, pp. 515–525, 2014.

[18] M. Barni, M. Fontani, and B. Tondi, "A universal technique to hide traces of histogram-based image manipulations," in *Proceedings of the on Multimedia and security*. ACM, 2012, pp. 97–104.

[19] ——, "A universal attack against histogram-based image forensics," *International Journal of Digital Crime and Forensics (IJDCF)*, vol. 5, no. 3, pp. 35–52, 2013.

[20] P. Comesana and F. Perez-Gonzalez, "The optimal attack to histogram-based forensic detectors is simple (x)," in *2014 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2014, pp. 137–142.

[21] C. Shi, L. Chen, C. Wang, X. Zhou, and Z. Qin, "Review of image forensic techniques based on deep learning," *Mathematics*, vol. 11, no. 14, 2023.

[22] E. Athanasiadou, Z. Geradts, and E. van Eijk, "Camera recognition with deep learning," *Forensic Sciences Research*, vol. 3, pp. 1–9, 10 2018.

[23] P. Sharma, M. Kumar, and H. Sharma, "Comprehensive analyses of image forgery detection methods from traditional to deep learning approaches: an evaluation," *Multimedia Tools and Applications*, vol. 82, pp. 1–34, 10 2022.

[24] S. Qureshi, A. Saeed, S. Almotiri, F. Ahmad, and M. Al Ghamdi, "Deepfake forensics: a survey of digital forensic methods for multimodal deepfake identification on social media," *PeerJ Computer Science*, vol. 10, p. e2037, 05 2024.

[25] M. Masood, M. Nawaz, K. M. Malik, A. Javed, and A. Irtaza, "Deepfakes generation and detection: State-of-the-art, open challenges, countermeasures, and way forward," 2021.

[26] T. T. Nguyen, Q. V. H. Nguyen, D. T. Nguyen, D. T. Nguyen, T. Huynh-The, S. Nahavandi, T. T. Nguyen, Q.-V. Pham, and C. M. Nguyen, "Deep learning for deepfakes creation and detection: A survey," *Computer Vision and Image Understanding*, vol. 223, p. 103525, Oct. 2022.

[27] B. U. Mahmud and A. Sharmin, "Deep insights of deepfake technology: A review," 2023.

[28] E. Nowroozi, A. Dehghantanha, R. M. Parizi, and K.-K. R. Choo, "A survey of machine learning techniques in adversarial image forensics," 2020.

[29] L. Fan, W. Li, and X. Cui, "Deepfake-image anti-forensics with adversarial examples attacks," *Future Internet*, vol. 13, no. 11, 2021.

[30] M. Barni, W. Li, B. Tondi, and B. Zhang, *Adversarial Examples in Image Forensics*. Singapore: Springer Singapore, 2022, pp. 435–466.

[31] D. Brunet, E. R. Vrscay, and Z. Wang, "On the mathematical properties of the structural similarity index," *Image Processing, IEEE Transactions on*, vol. 21, no. 4, pp. 1488–1499, 2012.

[32] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600–612, 2004.

[33] T. Pevnỳ and J. Fridrich, "Detection of double-compression in JPEG images for applications in steganography," *Information Forensics and Security, IEEE Transactions on*, vol. 3, no. 2, pp. 247–258, 2008.

[34] P. Comesana, L. Pérez-Freire, and F. Pérez-González, "Blind Newton sensitivity attack," in *Information Security, IEE Proceedings*, vol. 153, no. 3. IET, 2006, pp. 115–125.

[35] P. Comesana, F. Pérez-González, and F. Balado, "Optimal data-hiding strategies for games with BER payoffs," in *Digital Watermarking*. Springer, 2004, pp. 101–116.

[36] J. Nocedal and S. J. Wright, *Conjugate gradient methods*. Springer, 2006.

[37] T. E. Abrudan, J. Eriksson, and V. Koivunen, "Steepest descent algorithms for optimization under unitary matrix constraint," *Signal Processing, IEEE Transactions on*, vol. 56, no. 3, pp. 1134–1147, 2008.

[38] A. Bj orck, "Numerics of gram-schmidt orthogonalization," *Linear Algebra and its Applications*, vol. 197-198, pp. 297–316, 1994.

[39] N. J. Higham and R. S. Schreiber, "Fast polar decomposition of an arbitrary matrix," *SIAM J. Sci. Stat. Comput.*, vol. 11, no. 4, p. 648–655, Jul. 1990.

[40] 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.