This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2730822, IEEE Transactions on Information Forensics and Security

Localization of Diffusion-based Inpainting in Digital Images

Haodong Li, *Student Member, IEEE,* Weiqi Luo, *Senior Member, IEEE,* Jiwu Huang, *Fellow, IEEE*

***Abstract*—Image inpainting, an image processing technique for restoring missing or damaged image regions, can be utilized by forgers for removing objects in digital images. Since no obviously perceptible artifacts are left after inpainting, it is necessary to develop methods for detecting the presence of inpainting. In general, there are two main categories of image inpainting techniques: exemplar-based and diffusion-based techniques. Although several methods have been proposed for detecting exemplar-based inpainting, there is still no effective method for detecting diffusion-based inpainting. Usually, the tampered regions manipulated by diffusion-based inpainting techniques are much smaller than those manipulated by exemplar-based ones, presenting more challenges in detecting these regions. As a pioneering attempt, this paper proposes a method for the localization of diffusion-based inpainted regions in digital images. We first analyze the diffusion process in inpainting, and observe that the changes in the image Laplacian along the direction perpendicular to the gradient are different in the inpainted and untouched regions. Following this observation, we construct a feature set based on the intra-channel and inter-channel local variances of the changes to identify the inpainted regions. Finally, two effective post-processing operations are designed for further refining of the localization result. The extensive experimental results evaluated on both synthetic and realistic inpainted images show the effectiveness of the proposed method.**

***Index Terms*—Image forensics, Forgery localization, Image inpainting, Diffusion-based inpainting.**

I. Introduction

In recent years, the rapid development of image processing techniques and user-friendly software has facilitated the conveniences for retouching digital images. However, it has also led to an increase in image forgeries. Through the use of image processing tools, even an amateur forger can easily create a forged image with few visible traces. In order to address the harmful impacts introduced by maliciously altered images, image forensics [1], [2] have attracted considerable attention during the past decade. From a forensic point of view, any inherent traces introduced by the image generation pipeline or left by image processing operations can help in detecting image forgeries. For example, it is possible to perform camera source identification and further identify forgeries based on the sensor pattern noise (SPN) [3] and color filter array (CFA) properties [4], [5]; it is possible to detect image forgeries by analyzing distinct artifacts in JPEG compression [6], [7], contrast enhancement [8], [9], resampling [10], [11], and spatial filtering [12], [13]. If an image has inconsistent SPN or CFA patterns, or is subjected to image processing, it is very likely to be tampered with. Another approach for exposing image forgeries is to detect the presence of some typical tampering operations, such as splicing [14], [15] and copy-move [16], [17].

When using splicing or copy-move for the removal or insertion of an object in an image, the forger must carefully control the process to avoid leaving obvious artifacts such as inconsistent textures and/or disconnected boundaries. Thus, it is relatively hard for an amateur to create a perfect forgery. In the early 2000s, an image processing technique called *image inpainting* was developed; this is available within well-known image editing software such as Photoshop and GIMP. Image inpainting was originally designed for the restoration of missing or damaged regions in an image, imitating the work of art restoration workers. There are two main categories of image inpainting techniques: diffusion-based techniques [18]–[20] and exemplar-based techniques [21], [22]. With the help of image inpainting, one can modify an image using only a few clicks of the button. Although inpainting is not designed for falsification, a forger would probably use such a convenient technique to tamper with an image, such as removing some objects. Usually, exemplar-based inpainting techniques are suitable for removing relatively large objects. In many forgery applications, the targeted objects may be small, and thus the diffusion-based techniques, which leave very few perceptible artifacts in small areas, can also serve as powerful tampering tools. In practice, the credibility of an image must be suspected if it is detected as being inpainted, since inpainting would significantly change the contents and semantics. Therefore, it is valuable to develop forensic methods for the detection of image inpainting.

Hitherto, only a few works have focused on the forensics of image inpainting. Some of these have focused on detecting image inpainting in JPEG images, such as in [23] and [24]. The aim of these studies is to detect double JPEG compression; the inherent properties of image inpainting have not been well studied. Therefore, such methods still suffer from some tough problems in double JPEG compression detection, such

This work was supported in part by National Natural Science Foundation of China under Grant 61672551, and Grant U1636202, the Shenzhen R&D Program under Grant JCYJ20160328144421330, the Fok Ying-Tong Education Foundation under Grant 142003, the Special Research Plan of Guangdong Province under Grant 2015TQ01X365, and the Guangzhou Science and Technology Plan Project under Grant 201707010167.

H. Li is with the School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, China (e-mail: lihaod@mail2.sysu.edu.cn).

W. Luo (corresponding author) is with the School of Data and Computer Science, and Guangdong Key Laboratory of Information Security Technology, Sun Yat-sen University, Guangzhou 510006, China (e-mail: luoweiqi@mail.sysu.edu.cn).

J. Huang is with the College of Information Engineering and Shenzhen Key Laboratory of Media Security, Shenzhen University, Shenzhen 518052, China (e-mail: jwhuang@szu.edu.cn).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2730822, IEEE Transactions on Information Forensics and Security

(a) (b)

Fig. 1. An example of inpainting problem. (a) An image to be inpainted; the black region is unknown. (b) The inpainting result; the region surrounded by the dashed curve is recovered by inpainting.

as double compression with the same quantization matrix. Moreover, they are not useful when the inpainted image is not in JPEG format. The other methods for image inpainting detection, such as [25]–[29], are all designed to deal with the exemplar-based techniques. The common principle of these methods is to search similar blocks within the given image, similar to the block matching process in copy-move detection. The image block pairs with large matching degrees are then suspected to be inpainted ones. However, the diffusion-based inpainting techniques will not generate similar blocks in the inpainted regions, and thus the existing forensic methods for exemplar-based techniques cannot be adapted to detect diffusion-based inpainting. Moreover, the diffusion-based techniques are usually suitable for small areas and thus the size of inpainted region is rather small, which makes the detection of diffusion-based inpainting a difficult problem. So far, no effective method has been proposed for detecting diffusion-based inpainting.

In this paper, we propose a method for locating the tampered regions altered by diffusion-based inpainting techniques. We focus on analyzing the artifacts introduced by diffusion-based inpainting, which is an important and basic step for subsequent forensic analysis. By analyzing the basic properties of the diffusion process, we find that the changes in image Laplacians along the isophote directions in the inpainted regions are quite different from those in the untouched regions. Based on this property, we can extract certain features from such changes. The feature set consists of both intra-channel and inter-channel local variances of the changes computed with different window sizes. Then, we train a classifier to identify the inpainted pixels using the extracted features. In order to further improve the performance, we design two effective post-processing operations to refine the initial results: some abnormal exposed regions which lead to false alarms are excluded, and then morphological filtering with adaptive sizes of structuring elements is performed. Extensive experiments are conducted to evaluate the proposed method. The results obtained for 1000 synthetic inpainted images (the inpainted regions are randomly selected) and 200 realistic inpainted examples (the inpainted regions contain meaningful contents) show that the proposed method is effective for detecting diffusion-based inpainting. We also test the robustness of the proposed method to some commonly used post operations. The results show that the proposed method can still identify the inpainted regions when the images are subject to gamma correction / rotation / scaling, although producing more false alarms in some cases. For JPEG compression, the proposed method suffers from too many false alarms and thus the performance is poor, which needs to be further improved.

The rest of this paper is organized as follows. Section II gives a brief introduction to the image inpainting problem and diffusion-based techniques. Section III presents the details of the proposed method. Section IV reports and discusses the experimental results. Finally, concluding remarks are given in Section V.

II. PRELIMINARIES

A. The Inpainting Problem

Generally, an $N \times M$ image $\mathbf{I}$ can be regarded as a two-dimensional function defined on a domain D, that is

$$\mathbf{I} : D \to \mathbb{R}^m, \tag{1}$$

where $D = \{(x,y)|x = 0,1,\ldots,M; y = 0,1,\ldots,N\} \subset \mathbb{R}^2$ specifies the spatial coordinates of the image pixels, and m is the number of color channels ($m = 1$ for a gray image, and $m = 3$ for an RGB color image).

For the inpainting problem, an image $\mathbf{I}^0$ is given as shown in Fig. 1, in which the pixels in the black region are missing or contaminated[1]. Thus, $\exists \Omega \subset D$ such that for all $(x,y) \in \Omega$, the pixel values $\mathbf{I}^0(x,y)$ are unknown. Image inpainting aims to estimate (or recover) the pixel values in the unknown region Ω using only the information from the known region $S = D - \Omega$. As an ill-posed inverse problem, image inpainting does not have a unique solution. That is, the contents of

[1]It is noted that a very large inpainted region is chosen in this example for illustration purposes, so that the inpainted region is fairly blurred. In practice, the inpainted region is usually much smaller, and it is not generally easy to observe these blurred artifacts.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2730822, IEEE Transactions on Information Forensics and Security

the unknown region after inpainting will be different when different algorithms and parameters are applied. Nevertheless, the goal of inpainting must be fulfilled, i.e., the inpainted region is coherent with the known region and is visually acceptable. The general assumption in image inpainting is that both the known and unknown regions are consistent in terms of the structures of their geometry and/or have similar properties of textural statistics. The diffusion-based methods mainly exploit smoothness priors and partial differential equation (PDE) to propagate the structures from the outside to the inside of the unknown region, while the exemplar-based methods utilize image statistics and similarity priors to synthesize the patches in the unknown region. In this paper, we focus on diffusion-based inpainting and the basic idea of this technique is introduced in the following subsection.

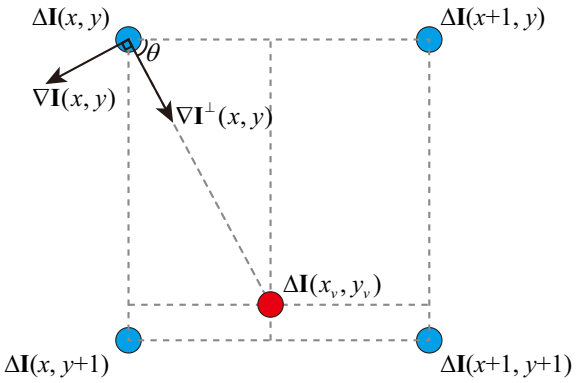


Fig. 2. Illustration of the calculation of the change in the image Laplacian along the isophote direction, where Δ is the Laplacian operator and ∇ is the gradient operator.

B. Diffusion-based Inpainting

The basic idea of diffusion-based inpainting is to propagate local information with smoothness constraints, like heat propagation in physical structures [30]. The propagation process imitates what is done by art restoration experts. In general, two issues need to be carefully considered in diffusion-based inpainting: the first is how to describe the local image structure, and the second is in which direction the local image structure should be propagated.

A typical diffusion-based inpainting method was proposed by Bertalmio et al. [18] in 2000. In this work, the image Laplacian is used as a smoothness predictor for describing the local structural information, and an anisotropic model is employed to propagate the image Laplacian along the direction of the image isophote, which is perpendicular to the image gradient in each pixel point. Formally, the algorithm updates the pixel intensities iteratively inside the unknown region by solving the following equation,

$$\mathbf{I}^{t+1}(x,y) = \mathbf{I}^{t}(x,y) + t' \cdot \mathrm{d}\mathbf{I}^{t}(x,y), \forall (x,y) \in \Omega, \tag{2}$$

where t is the iteration time, t' is the update speed, and $\mathrm{d}\mathbf{I}^{t}(x,y)$ is the update signal for $\mathbf{I}^{t}(x,y)$. $\mathrm{d}\mathbf{I}^{t}(x,y)$ is given by

$$\mathrm{d}\mathbf{I}^{t}(x,y) = \nabla(\Delta\mathbf{I}^{t}(x,y)) \cdot \nabla\mathbf{I}^{t\perp}(x,y), \tag{3}$$

where ∇ is the gradient operator, $\Delta\mathbf{I}$ represents the image Laplacians, and $\nabla\mathbf{I}^{\perp}$ is the isophote direction (perpendicular to the gradient direction). The term $\nabla(\Delta\mathbf{I}) \cdot \nabla\mathbf{I}^{\perp}$ represents the derivative of $\Delta\mathbf{I}$ in the isophote direction. In the initial state, $t = 0$ and $\mathbf{I}^{t} = \mathbf{I}^{0}$. After several iterations, a convergent status is obtained such that $\mathrm{d}\mathbf{I}^{t} = 0$, and the resulting inpainted image is defined as the output of the convergent status. After convergence, there is no variation in the image Laplacians $\Delta\mathbf{I}$ in the directions of the isophote $\nabla\mathbf{I}^{\perp}$, meaning that the image information (i.e., the Laplacians) is propagated inside the unknown region in a way that aims to preserve the isophote directions.

Diffusion-based inpainting methods tend to prolong structures (e.g., isophotes [18] and curvatures [20]) arriving at the boundary of the region to be filled in; hence they are suitable for propagating strong structures, or for filling small regions. If the region to be completed is large or of complex texture, the inpainted region appears blurred after a few diffusion iterations. Therefore, when such a technique is used for object removal, the altered region tends to be small, otherwise there would be obviously visible artifacts. For this reason, we focus on detecting small inpainted regions in this work.

III. The Proposed Method

In this section, we first point out a typical artifact of diffusion-based inpainting, and then construct a feature set to identify the inpainted regions and design certain post-processing operations to refine the localization results.

A. Artifact of Diffusion-based Inpainting

In diffusion-based inpainting, the central concept is to solve the PDEs for a solution of the unknown regions. Therefore, the inpainted regions will to some extent suffer from the blurring effects introduced by the diffusion process. Fig. 1 (b) shows the typical blurring effects of diffusion-based inpainting, where we apply inpainting to an extra-large region for illustration purposes. In fact, such blurring effects also appear when the inpainted region is small, but are not easily visible to the human eye. It seems that the blurring artifacts can help to expose diffusion-based inpainting. However, due to the diversity of natural image contents, an original image that has never been inpainted may also contain some smooth regions, leading to difficulties in effectively detecting and locating the inpainted regions. In order to differentiate the inpainted regions from the untouched ones, therefore, we need to analyze the blurring artifacts left by the diffusion process and investigate how such artifacts differ from those in the untouched regions.

We take the inpainting algorithm proposed in [18] as a typical example for analysis. Based on Eq. (2) and (3), we know that the pixels in the inpainted region will satisfy $\mathrm{d}\mathbf{I}^{t} = 0$, meaning that the image Laplacians will remain constant in the directions of the isophote. This property, which is expected not to be held in the untouched regions, leads to the blurring artifacts in the inpainted regions. To evaluate whether this property can differentiate inpainted and untouched regions, we first calculate the change in the image Laplacian along

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2730822, IEEE Transactions on Information Forensics and Security

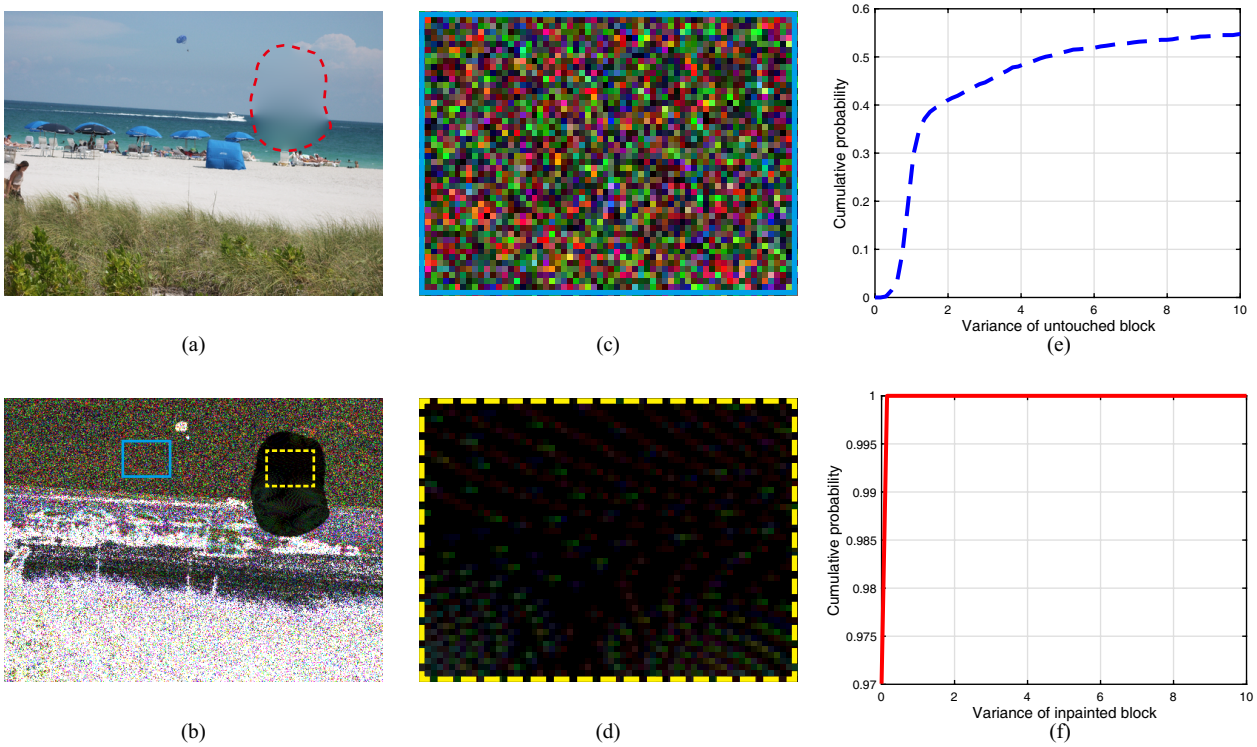


Fig. 3. Illustration of the changes in image Laplacians along the isophote directions. (a) An inpainted image with the inpainted region surrounded by the dashed curve. (b) The map of $|\delta_{\Delta\mathbf{I}}|$. (c) Illustration of the cyan block in (b) with a six-fold enlargement. (d) Illustration of the yellow dashed block in (b) with a six-fold enlargement. (e) CDF of the variances of untouched blocks. (f) CDF of the variances of inpainted blocks.

the direction of the isophote as illustrated in Fig. 2. For each image pixel, we compute

$$\delta_{\Delta\mathbf{I}}(x,y) = \Delta\mathbf{I}(x,y) - \Delta\mathbf{I}(x_v,y_v), \forall (x,y) \in D, \tag{4}$$

where $\Delta\mathbf{I}(x,y)$ is the value of the image Laplacian for the pixel with coordinate (x,y), and $\Delta\mathbf{I}(x_v,y_v)$ is the value of the image Laplacian for a virtual pixel point. The virtual pixel is located at the direction of $\nabla\mathbf{I}^{\perp}(x,y)$, and its distance between the pixel $\mathbf{I}(x,y)$ is 1 (see the red point in Fig. 2). The coordinate (x_v,y_v) is given by

$$\begin{pmatrix} x_v \\ y_v \end{pmatrix} = \begin{pmatrix} x+\cos\theta \\ y+\sin\theta \end{pmatrix}, \tag{5}$$

where $\tan\theta = |\nabla\mathbf{I}^{\perp}{}_y(x,y)|/|\nabla\mathbf{I}^{\perp}{}_x(x,y)|$, and $\nabla\mathbf{I}^{\perp}{}_x(x,y)$, $\nabla\mathbf{I}^{\perp}{}_y(x,y)$ are the projections of the isophote over the horizontal and vertical directions, respectively. The value of $\Delta\mathbf{I}(x_v,y_v)$ is computed with bilinear interpolation as follows,

$$\Delta\mathbf{I}(x_v,y_v) = \begin{pmatrix} 1-\cos\theta \\ \cos\theta \end{pmatrix}^T N_{\Delta\mathbf{I}} \begin{pmatrix} 1-\sin\theta \\ \sin\theta \end{pmatrix}, \tag{6}$$

where $N_{\Delta\mathbf{I}}$ is given by

$$N_{\Delta\mathbf{I}} = \begin{pmatrix} \Delta\mathbf{I}(x,y) & \Delta\mathbf{I}(x+1,y) \\ \Delta\mathbf{I}(x,y+1) & \Delta\mathbf{I}(x+1,y+1) \end{pmatrix}. \tag{7}$$

By applying Eq. (4) to a given image, we can obtain a map $\delta_{\Delta\mathbf{I}}$, whose absolute values are as shown in Fig. 3 (b). From Fig. 3 (b), we can observe that the patterns in the inpainted region and the untouched region are quite different. In the untouched region, the changes of the image Laplacians along the isophote directions have large intensities, while the changes in the inpainted region are of low intensities. Such phenomena can be clearly observed in two enlarged parts, as shown in Fig. 3 (c) and (d). From these two sub-figures, we can further find out that the range (i.e., the difference between the maximum and minimum values) of the intensities for a local area in the untouched region is large, while the intensity range in the inpainted region is relatively small, indicating that the local variances in the inpainted region should be smaller than those in the untouched regions. In order to quantitatively investigate the properties of the map $\delta_{\Delta\mathbf{I}}$, we compute the local variance for each 8×8 non-overlapping image block, and draw the curves of the empirical cumulative probability function (CDF) for the variances of untouched and inpainted blocks, respectively. The resulting CDF curves are shown in Fig. 3 (e) and (f), from which we can observe that the variances of the inpainted blocks are close to zero, much less than those of the untouched blocks.

Based on the above analysis, we conclude that the changes in the image Laplacians along the direction of isophote (i.e., the map $\delta_{\Delta\mathbf{I}}$) can indeed capture the blurring artifacts left by diffusion-based inpainting, and thus the main idea of the proposed method is to use such an evidence to detect diffusion-based inpainting.

B. Feature Extraction and Classification

In the above subsection, we have shown that the map $\delta_{\Delta\mathbf{I}}$ can help in distinguishing between inpainted and untouched regions. In order to perform automatic and accurate localiza-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2730822, IEEE Transactions on Information Forensics and Security

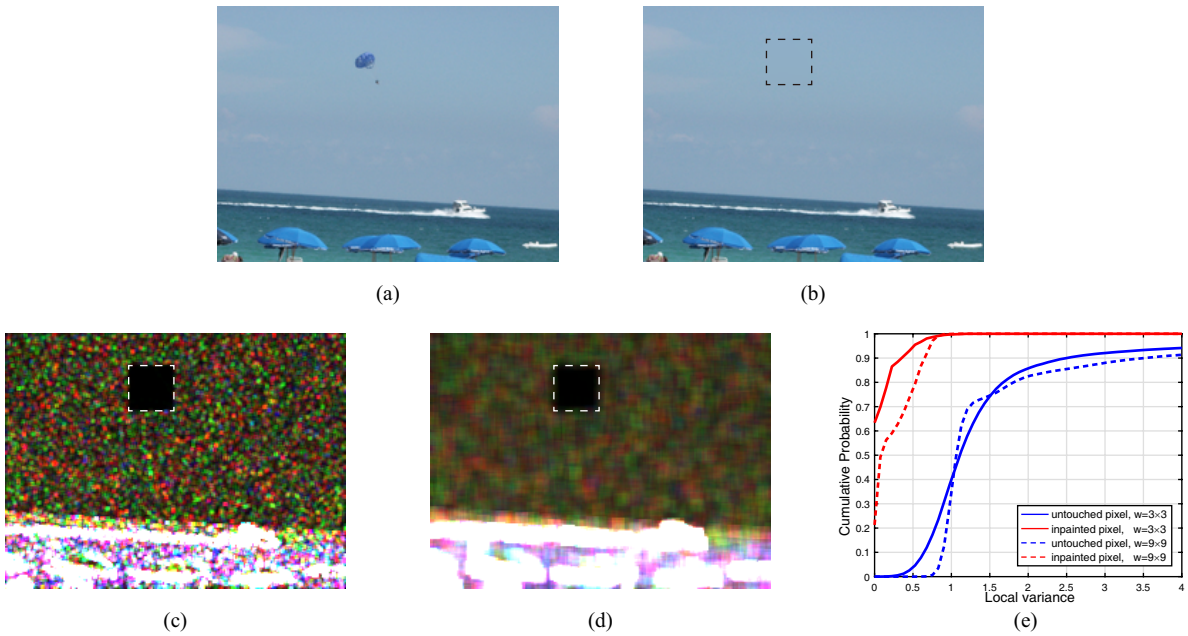


Fig. 4. The local variances of the map $\delta_{\Delta\mathbf{I}}$ with different window sizes. (a) Original image. (b) Inpainted image with the inpainted region surrounded by the dashed square. (c) Local variances computed by window size $w = 3 \times 3$. (d) Local variances computed by window size $w = 9 \times 9$. (e) The empirical CDFs of the local variances of untouched/inpainted pixels with two different window sizes.

tion, we try to extract some statistical features from $\delta_{\Delta\mathbf{I}}$ and perform classification.

Since the previous observations and analysis indicate that the local variance would be a kind of effective feature, we can compute the local variances of $\delta_{\Delta\mathbf{I}}$ with a sliding window when giving a suspected image (as shown in Fig. 4 (b)). As a result, we obtain the maps of local variances as shown in Fig. 4 (c) and (d), from which we can claim that those pixels associated with small local variances are inpainted ones. However, a tricky problem is how to determine the size of the sliding window. From Fig. 4 (c), we observe that a small window size ($w = 3 \times 3$) introduces some local anomalies in the untouched region. That is, some untouched pixels also produce relatively small variances; this will lead to false alarms. On the other hand, Fig. 4 (d) shows that a large window size ($w = 9 \times 9$) results in relatively large variances at the boundary of the inpainted region; this will lead to missed detections especially when the inpainted region is small. These two phenomena are reflected in the curve of the empirical cumulative probability function (CDF) as shown in Fig. 4 (e). It is therefore shown that neither the result of $w = 9 \times 9$ nor that of $w = 3 \times 3$ is superior to the other. As a compromise, we use different window sizes to compute the local variance of $\delta_{\Delta\mathbf{I}}$. It is noted that we should not use a very large window, because the inpainted regions in an image (if exist) are usually small and a too large window size would confuse the local variances in the inpainted regions with those in the untouched regions.

Given an RGB color image, we firstly compute the *intra-channel local variance* of $\delta_{\Delta\mathbf{I}}$. For each color channel $c \in \{1,2,3\}$ and each pixel coordinate (x,y), the intra-channel local variance with window size w is denoted as $\sigma^2_{c,w}$, which is calculated as

$$\sigma^2_{c,w}(x,y) = \frac{1}{w^2} \sum_{i=-\lfloor \frac{w}{2} \rfloor}^{\lfloor \frac{w}{2} \rfloor} \sum_{j=-\lfloor \frac{w}{2} \rfloor}^{\lfloor \frac{w}{2} \rfloor} \left(\delta_{\Delta\mathbf{I}c}(x+i, y+j) - \mu_c(x,y) \right)^2, \quad (8)$$

where $\lfloor \cdot \rfloor$ is the rounding operator that rounds a number to its nearest integer towards negative infinity, $\delta_{\Delta\mathbf{I}c}$ is the c-th component of $\delta_{\Delta\mathbf{I}}$, and μ_c is the mean value of $\delta_{\Delta\mathbf{I}c}$ within the $w \times w$ window. In the experiments, we choose $w \in \{3,5,7,9\}$, producing a 12 (=3×4) dimensional feature for each pixel.

In addition to intra-channel variance, we also consider the *inter-channel local variance* ς^2_w, which is computed as

$$\varsigma^2_w(x,y) = \frac{1}{3w^2} \sum_{c=1}^{3} \sum_{i=-\lfloor \frac{w}{2} \rfloor}^{\lfloor \frac{w}{2} \rfloor} \sum_{j=-\lfloor \frac{w}{2} \rfloor}^{\lfloor \frac{w}{2} \rfloor} \left(\delta_{\Delta\mathbf{I}c}(x+i, y+j) - \mu(x,y) \right)^2, \quad (9)$$

where μ is the mean value of all three components of $\delta_{\Delta\mathbf{I}}$ within the $w \times w$ window, i.e., $\mu(x,y) = \frac{1}{3} \sum_{c=1}^{3} \mu_c(x,y)$. In the experiments, we choose $w \in \{1,3,5,7\}$ to produce a 4-dimensional feature for each pixel, and thus construct a 16-dimensional intra-channel and inter-channel local variance feature.

At this point, we can collect training samples from some pre-labelled inpainted images. For each training image, we extract the 16-D features from the untouched pixels as negative samples and the inpainted pixels as positive samples, respectively. Then, the positive and negative samples are used to train a classifier. For an image under investigation, we firstly extract the features from each of its pixel, and then feed the features to the trained classifier to predict whether a pixel is inpainted or not. Finally, we generate a binary localization map $\mathbf{M}$ with the same size of the given image, where the pixels predicted as inpainted are labelled with the value 1 and the others are labelled with 0.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2730822, IEEE Transactions on Information Forensics and Security

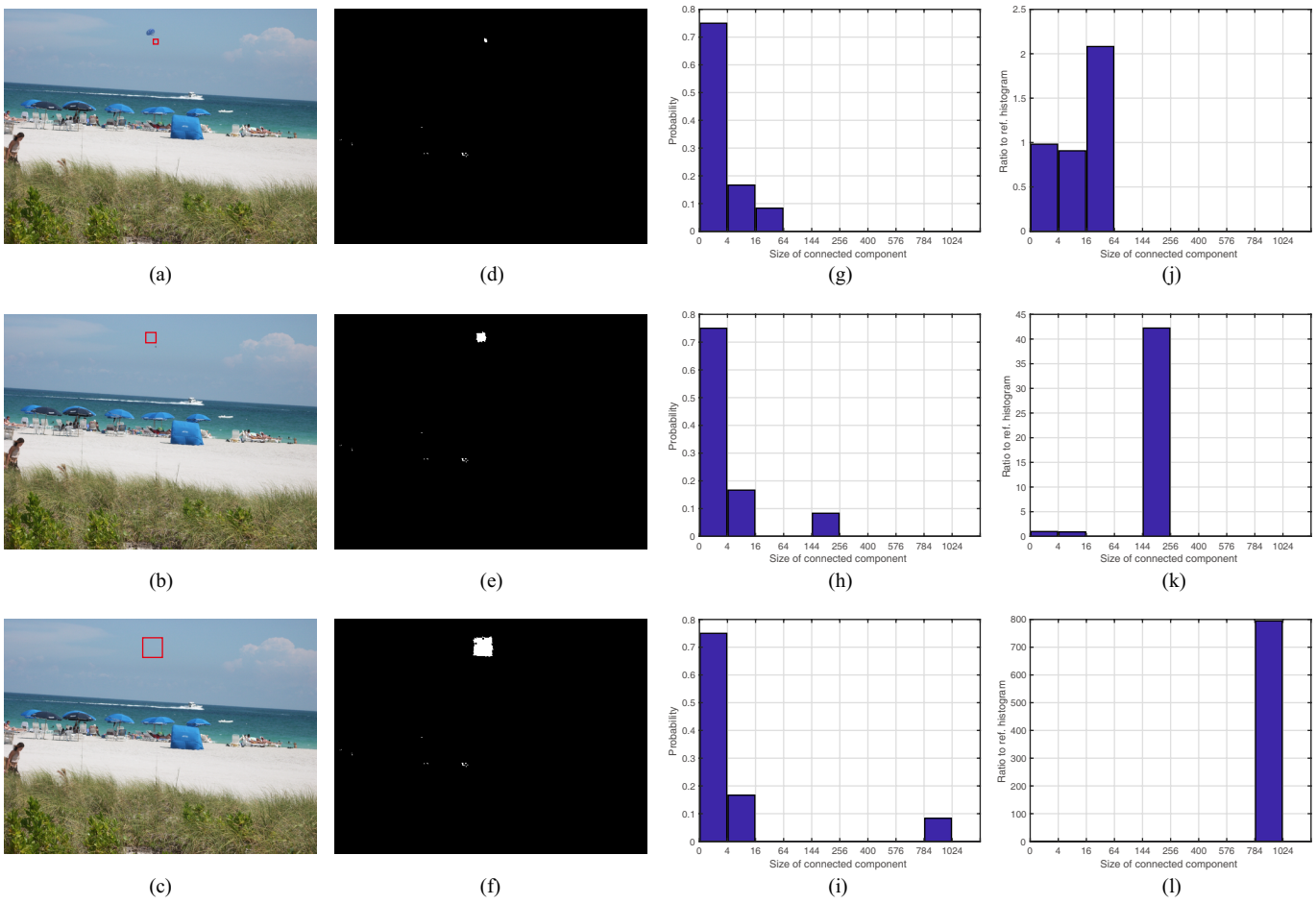


Fig. 5. (a)-(c) Three images with 8×8, 16×16, and 32×32 inpainted regions, respectively. (d)-(f) The corresponding localization maps. (g)-(i) Histograms of the size of connected components in the localization maps. (j)-(l) The ratios of the histograms to the referenced histograms.

C. Post-Processing

Although we have obtained a localization map as described in Section III-B, the result is not so satisfactory since there may be some false alarms or missed detections. Therefore, we further perform two operations to refine the localization results, that is, the exclusion of abnormal exposed regions and morphological filtering.

1) Exclusion of abnormal exposed regions: Sometimes, there may be underexposed and overexposed regions in an image, which are presented as very black and very bright regions, respectively. In such regions, the values of image Laplacians are close to zeros. Thus, the local variances of $\delta_{\Delta \mathbf{I}}$ will be very small. In this case, the pixels within such regions will be classified as inpainted ones. In order to re-predict such pixels as untouched ones, we define a measure of abnormal exposure E:

$$\begin{aligned} E(x,y) = &\sum_{c=1}^{3}\sum_{i=-1}^{1}\sum_{j=-1}^{1} \mathbb{1}\big(\mathbf{I}_c(x+i,y+j) < 10\big) \\ &+ \sum_{c=1}^{3}\sum_{i=-1}^{1}\sum_{j=-1}^{1} \mathbb{1}\big(\mathbf{I}_c(x+j,y+j) > 245\big), \end{aligned} \tag{10}$$

where $\mathbb{1}(\cdot)$ is an indicator function, and $\mathbf{I}_c(x,y)$ is the c-th component of the pixel intensity at coordinate (x,y). For pixels with $E(x,y) = 27$, that is, the values of the RGB components of their 3×3 neighbors and themselves are all less than 10 or larger than 245, we treat them as abnormal exposed pixels and predict them as untouched ones. In this way, the underexposed and overexposed regions are avoid being erroneously detected as inpainted regions.

2) Morphological filtering: Another operation to improve the localization performance is morphological filtering. We firstly perform erosion to remove some small false-alarmed regions, and then apply dilation to enlarge the regions detected as inpainted so that the true positive results will be increased. The process of morphological filtering can be described as the following equation,

$$\hat{\mathbf{M}} = \big(\mathbf{M} \ominus S(r_e)\big) \oplus S(r_d) \tag{11}$$

where $\hat{\mathbf{M}}$ is the morphological filtered localization map, $\mathbf{M}$ is the input localization map, and $S(r_e)$ and $S(r_d)$ are disk-shaped structuring elements with radius r_e and r_d, respectively. It is noted that the size of structuring element will strongly affect the performance. If the inpainted region is small, we need to use a small structuring element to avoid the removal of the actually inpainted region. If the inpainted region is large, we prefer to use a large structuring element to eliminate more false-alarmed pixels. Since the size of the inpainted region

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2730822, IEEE Transactions on Information Forensics and Security

is unknown, we design a method to estimate its size based on the localization map. We first compute the histogram of the size of *connected components*[2] in the localization map. The histogram H is represented with 10 bins, as shown in Fig. 5 (g)-(i). According to Fig. 5 (d)-(f), the false-alarmed regions are usually of small sizes and do not connect to each other. Thus, most of the false-alarmed regions contribute to the bins representing small sizes in the histogram, while the inpainted regions are associated to the bins representing large sizes. Through the use of training images, we can pre-compute a referenced histogram H^r that only counts the size of false-alarmed connected components. For a given image, we compute its histogram of the size of connected components in the localization map, and obtain the ratio to the referenced histogram for each bin as $R(=H/H^r)$, as shown in Fig. 5 (j)-(l). At this point, the sizes of inpainted regions are related to the positions of local maxima in R, which are denoted as

$$\Lambda = \big\{\lambda \big| R(\lambda) > R(\lambda-1), \text{ and } R(\lambda) > R(\lambda+1), \text{ and } \lambda \in \{2,3,\ldots,9\}\big\}, \tag{12}$$

where $R(\lambda)$ is the λ-th element of R. In the case that there are several inpainted regions with different sizes, we should select the sizes of structuring elements based on the smaller one, so as to avoid using a big structuring element which may erase the smaller inpainted regions. Therefore, we determine the sizes of structuring elements based on the first local maximum position in the bin-to-bin ratio R, i.e., $\min\{\Lambda\}$.

IV. Experimental Results

In the experiments, we employed the proposed method to detect three diffusion-based inpainting algorithms (isotropic, edge-oriented, and Delaunay-oriented) provided by G'MIC [31], a full-featured open-source framework for image processing. Although these algorithms exploit different models to describe the structural information for diffusion, they are all based on the fundamental concepts proposed in [18].

Our objective is to detect and locate the inpainted regions in an image. The performance is measured by the F1-score

$$\mathrm{F1} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FN} + \mathrm{FP}}, \tag{13}$$

where TP (true positive), FN (false negative), and FP (false positive) mean the number of detected inpainted pixels, undetected inpainted pixels, and wrongly detected untouched pixels, respectively.

A. Performance for Synthetic Inpainted images

In this experiment, we evaluated the localization performance for synthetic inpainted images. We adopted the 1338 color images in the UCID database [32] with sizes of 384×512 or 512×384. 338 images are randomly selected for training the classifier, as mentioned in Section III-B, and for determining the sizes of structuring elements, as mentioned in Section III-C. The rest of the images were used for testing. For each image in the database, we randomly inpainted one region within it using the three inpainting algorithms with three different shapes and four different sizes. The inpainting shapes are square, circular, and irregular (a solid region with irregular shape). The inpainting sizes are 64-pixel(px), 256-px, 1024-px, and 4096-px, which are 0.33‰, 1.30‰, 5.21‰, and 2.08% of the whole image size, respectively. As a result, we obtain 36 inpainted images from each original image.

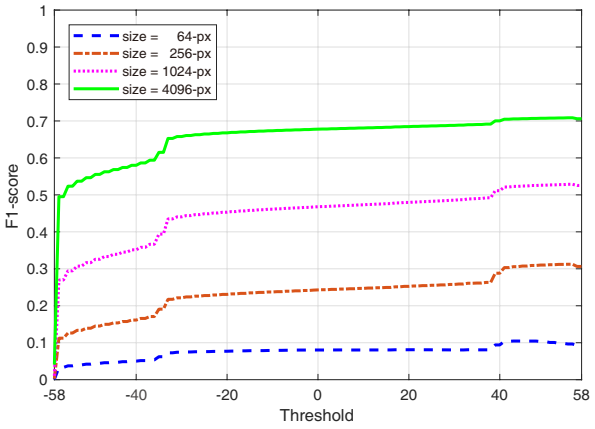


Fig. 6. The F1-score vs. threshold for the sum of ensemble votes.

For training the classifier, we first extracted the 16-D feature from each pixel in a training image. Since the training process is very time-consuming when using all the samples, we kept only 50 positive samples[3] and 50 negative samples for each image. Hence, we obtained totally 608400 (=$338\times36\times50$) positive samples and 608400 negative samples. We then trained an ensemble classifier [33] with linear discriminant analysis base learners using these samples. The resulting ensemble classifier consisted of 58 base learners, each of which output either a vote $+1$ (predicted as positive) or -1 (predicted as negative). The sum of all votes was then compared to a threshold for predicting the class of the input pixel. In order to optimize the performance based on F1-score, we conducted some preliminary experiments to determine the threshold. Fig. 6 shows the F1-scores obtained by different thresholds (please note that the F1-scores here were obtained without the post-processing operations introduced in Section III-C), from which we observe that the performance is improved by increasing the threshold and the highest average F1-score is obtained with the threshold of 50. Therefore, we choose 50 as the threshold for the sum of ensemble votes, i.e., a pixel is predicted as inpainted when the sum of the votes is larger than or equal to 50.

By using the trained classifier, we obtained an initial localization map for each training image. Subsequently, we excluded the abnormal exposed regions in the localization maps as described in Section III-C (a). And then, we computed a referenced histogram based on the false-alarmed connected components. By comparing the histograms of connected com-

[2]A connected component in the localization map is a set of pixels with value 1, where each pixel has at least one 8-adjacent neighbor and its 8-adjacent neighbors are all in the same set.

[3]The maximum number of positive samples in an image vary from 64 to 4096 according to the inpainting size. In order to avoid bias for inpainting size in the training process, we use the same samples (i.e., 50) for every image.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2730822, IEEE Transactions on Information Forensics and Security

TABLE I
F1-SCORES OBTAINED ON THE UCID DATABASE FOR DIFFERENT INPAINTING ALGORITHMS, INPAINTING SHAPES AND SIZES.

Algorithm	Isotropic			Edge-oriented			Delaunay-oriented		
Shape	square	circular	irregular	square	circular	irregular	square	circular	irregular
Size = 4096-px	0.9015	0.8980	0.8861	0.8938	0.8906	0.8753	0.8211	0.8714	0.8350
Size = 1024-px	0.8131	0.8097	0.7831	0.8023	0.7978	0.7654	0.6194	0.7038	0.6815
Size = 256-px	0.6964	0.6961	0.6533	0.6863	0.6856	0.6302	0.3359	0.3399	0.4042
Size = 64-px	0.4248	0.5230	0.1831	0.3339	0.4389	0.1582	0.0975	0.1262	0.0763

TABLE II
THE DEGRADATIONS OF F1-SCORES WITHOUT MORPHOLOGICAL FILTERING.

Algorithm	Isotropic			Edge-oriented			Delaunay-oriented		
Shape	square	circular	irregular	square	circular	irregular	square	circular	irregular
Size = 4096-px	↓0.0414	↓0.0331	↓0.0497	↓0.0481	↓0.0421	↓0.0561	↓0.0704	↓0.0594	↓0.0649
Size = 1024-px	↓0.0856	↓0.0744	↓0.0931	↓0.0913	↓0.0819	↓0.0956	↓0.1117	↓0.1098	↓0.1067
Size = 256-px	↓0.1727	↓0.1636	↓0.1916	↓0.1783	↓0.1712	↓0.1879	↓0.1283	↓0.1219	↓0.1473
Size = 64-px	↓0.2150	↓0.2486	↓0.0883	↓0.1677	↓0.2043	↓0.0762	↓0.0483	↓0.0621	↓0.0382

TABLE III
THE DEGRADATIONS OF F1-SCORES WITHOUT EXCLUSION OF ABNORMAL EXPOSED REGIONS AND MORPHOLOGICAL FILTERING.

Algorithm	Isotropic			Edge-oriented			Delaunay-oriented		
Shape	square	circular	irregular	square	circular	irregular	square	circular	irregular
Size = 4096-px	↓0.1423	↓0.1348	↓0.1487	↓0.1454	↓0.1396	↓0.1508	↓0.1627	↓0.1576	↓0.1585
Size = 1024-px	↓0.2215	↓0.2122	↓0.2225	↓0.2231	↓0.2151	↓0.2201	↓0.2102	↓0.2248	↓0.2160
Size = 256-px	↓0.2999	↓0.2929	↓0.3034	↓0.3008	↓0.2952	↓0.2944	↓0.1735	↓0.1700	↓0.2058
Size = 64-px	↓0.2704	↓0.3234	↓0.1109	↓0.2110	↓0.2656	↓0.0946	↓0.0581	↓0.0748	↓0.0450

ponents in the localization maps for the training images to the referenced histogram, we found that the 64-px and 256-px inpainted regions respectively produce the first local maximum in the bin-to-bin ratio at the 3rd and 5th bins (i.e., $\min\{\Lambda\} = 3$ and $\min\{\Lambda\} = 5$), while the 1024-px and 4096-px inpainted regions usually lead to $\min\{\Lambda\} \geq 8$. Hence, we set three different sizes for the structuring elements. Via searching the optimal sizes on the training images, for an image with $\min\{\Lambda\} \leq 3$, we set $r_e = 1$ and $r_d = 4$; for an image with $3 < \min\{\Lambda\} \leq 5$, we set $r_e = 2$ and $r_d = 5$; for the rest of the cases, we set $r_e = 4$ and $r_d = 6$.

Based on the trained classifier and the pre-set parameters at hand, we detected and located the inpainted regions in the testing images. For each inpainting algorithm, and each shape and size of the inpainted region, 1000 testing images were analyzed. The false positive (false alarm) rates for different cases ranges from 0.73% to 0.85%, and the average false positive rate is 0.78%, meaning that only a very small proportion of the untouched pixels are mistakenly detected as inpainted ones. The resulting F1-scores for different cases are shown in Table I. We observe that the size of the inpainted region is the main factor influencing the localization performance. Fig. 7 shows the box plots of F1-scores for different inpainting sizes, from which we can observe that the best F1-scores are close to 1 and the worst ones are 0, while the average F1-score becomes larger when the inpainting size increases. For 4096-px inpainted regions, we can achieve the average F1-score of 0.8748, meaning that the proposed method can effectively locate the inpainted regions. However, we merely obtain the average F1-score of 0.2624 for 64-px inpainted regions, because the inpainting size is very small so that much fewer tampering artifacts are left. We note that even the F1-score is lower than 0.3 in this case, it is still possible to figure out the inpainted region based on the localization map. Please refer to the realistic examples shown in Fig. 8 and Fig. 9. Regarding different inpainting shapes, the average F1-scores for square, circular, and irregular shapes are 0.6188, 0.6484, and 0.5776, respectively. The irregular shape degrades the performance, but the influence is not as significant as for different inpainting sizes. In comparing the results for different inpainting algorithms, it is shown that the performance for the Delaunay-oriented method is relatively poor. Nevertheless, we can achieve the average F1-score of 0.6682 when the inpainting size is 1024-px, meaning that in this case the inpainted region can still be satisfactorily identified by the proposed method.

In order to evaluate the necessity of the application of post-processing, we tested the performance of two variants of the proposed method, i.e., without morphological filtering and without any post-processing. Under the test conditions described above, the performance degradations of the two variants compared to the proposed method are shown in Tables II and III, respectively. We can observe that the localization performance is distinctly degraded when the post-processing operations are omitted. On average, omitting the morphological filtering leads to a decrement of 0.11 for the F1-score,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2730822, IEEE Transactions on Information Forensics and Security

TABLE IV
F1-SCORES FOR THE EXEMPLAR-BASED INPAINTING METHOD IN [26] OBTAINED ON THE UCID DATABASE FOR DIFFERENT INPAINTING ALGORITHMS, INPAINTING SHAPES AND SIZES.

Algorithm	Isotropic			Edge-oriented			Delaunay-oriented		
Shape	square	circular	irregular	square	circular	irregular	square	circular	irregular
Size = 4096-px	0.0261	0.0208	0.0209	0.0325	0.0278	0.0237	0.0133	0.0137	0.0139
Size = 1024-px	0.0201	0.0160	0.0157	0.0147	0.0094	0.0112	0.0073	0.0079	0.0080
Size = 256-px	0.0028	0.0018	0.0018	0.0023	0.0017	0.0022	0.0009	0.0009	0.0012
Size = 64-px	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001

TABLE V
F1-SCORES FOR THE MEDIAN FILTERING DETECTION METHOD IN [13] OBTAINED ON THE UCID DATABASE FOR DIFFERENT INPAINTING ALGORITHMS, INPAINTING SHAPES AND SIZES.

Algorithm	Isotropic			Edge-oriented			Delaunay-oriented		
Shape	square	circular	irregular	square	circular	irregular	square	circular	irregular
Size = 4096-px	0.7377	0.7369	0.7156	0.7295	0.7270	0.7053	0.7582	0.6936	0.7036
Size = 1024-px	0.5475	0.5441	0.5219	0.5404	0.5307	0.5076	0.5828	0.4733	0.4805
Size = 256-px	0.3123	0.3162	0.2956	0.3079	0.3052	0.2821	0.3744	0.2915	0.2545
Size = 64-px	0.0919	0.1253	0.0870	0.0799	0.1127	0.0752	0.1260	0.1312	0.0531

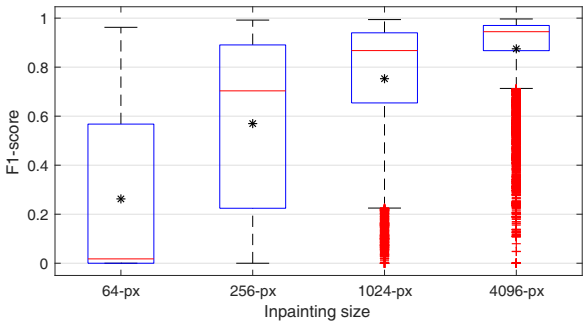


Fig. 7. Box plots of F1-scores for different inpainting sizes. The black asterisks represent the mean values.

while the omission of both post-processing operations leads to an additional decrement of 0.09. With regard to the three sizes of inpainted region, the performance for the 256-px region is influenced most obviously by the post-processing operations. The average F1-score drops 0.16 when the morphological filtering is left out and drops 0.26 when no post-processing is applied.

At last, we compared the performance of the proposed method to two related methods: the exemplar-based inpainting detection method [26] and the median filtering detection method [13]. The default parameters reported in [26] and [13] were used, except that a 8×8 sliding window with a step of 4-px was used with the method [13] since the smallest inpainted region was 64-px (equivalent to 8×8-px). The F1-scores obtained with the two methods are shown in Tables IV and V, respectively. From Table IV, we can observe that the method in [26], designed for the detection of exemplar-based inpainting, totally fails to detect diffusion-based inpainting, where the highest F1-score is only 0.0325. As the inpainted regions present some blurring artifacts, it is expected that median filtering detection methods (such as [13]) can identify them more or less. However, since the filtering detection method is not specially designed for diffusion-based inpainting, its performance may not be satisfactory. From Table V we can observe that the method in [13] can achieve the performance comparable with the proposed method for Delaunay-oriented inpainting when the inpainting size is less than 1024-px, while its performance for other cases are obviously poorer. Such results indicate that the existing filtering detection methods are still not suitable for detecting and locating the diffusion-based inpainted regions.

B. Performance for Realistic Forgeries

In this experiment, we evaluated the proposed method with some realistic inpainted images. We chose 100 images from the UCID database and 100 images the BossBase database [34]. The images in BossBase were originally stored in raw format and with sizes ranging from 2000×3008 to 5212×3468, which were converted into RGB TIFF images in this experiment. Within each of the 200 images, we selected one or several objects for removal with inpainting. All of the selected objects contained meaningful content, for example a flower on the ground, the window of a house, and so on. Since diffusion-based inpainting is suitable for small regions, the largest size of the selected objects was about 1% of the whole image size. We respectively used the three inpainting algorithms to inpaint the selected regions, and thus generated 600 inpainted images. In the first rows of Fig. 8 and Fig. 9, we show some examples of the inpainted images. We can observe that it is hard to identify the inpainted regions by human eye.

We detected the inpainted images with the same classifier and parameters obtained from the training set of UCID synthetic images (used in the previous experiment). The resulting false positive rate is also very low. The average false positive rate is 0.35% for the images from UCID, and 1.59% for the images from BossBase. The false positive rate for BossBase is larger than that for UCID. The images from BossBase have higher resolution and higher quality, and thus less noise is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2730822, IEEE Transactions on Information Forensics and Security

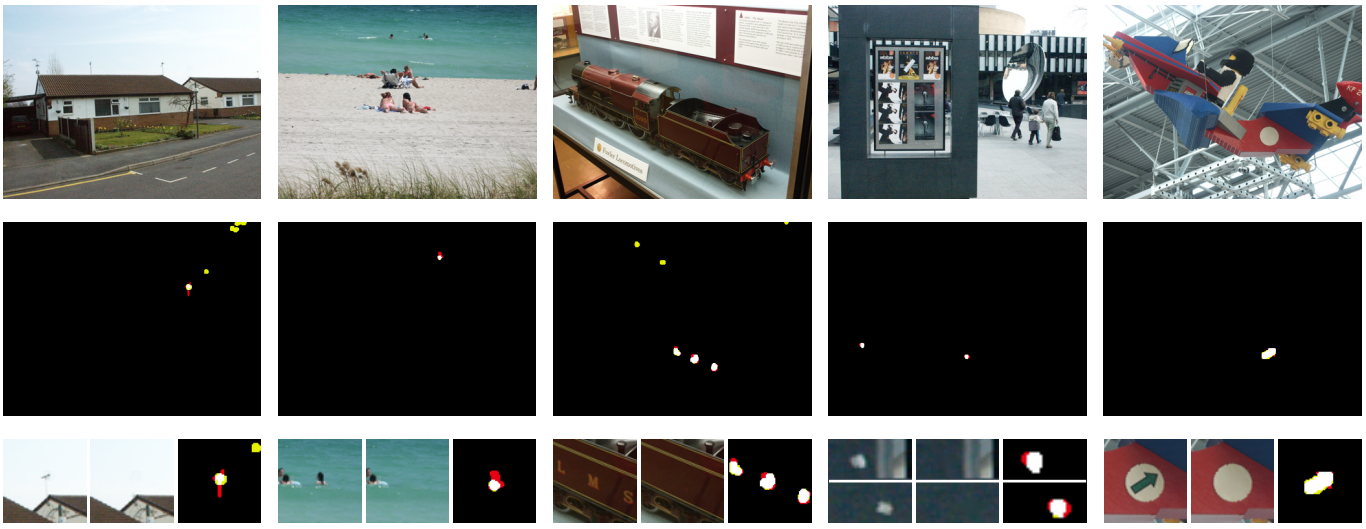


Fig. 8. Examples from the UCID database. Row 1: inpainted images. Row 2: localization results, where the pixels in white, black, yellow, and red indicate true positive, true negative, false positive, and false negative, respectively. Row 3: the enlarged parts of the ordinal regions, the inpainted regions, and the localization results. The F1-scores for these examples are 0.2348, 0.5792, 0.7064, 0.8750, 0.9059, respectively.

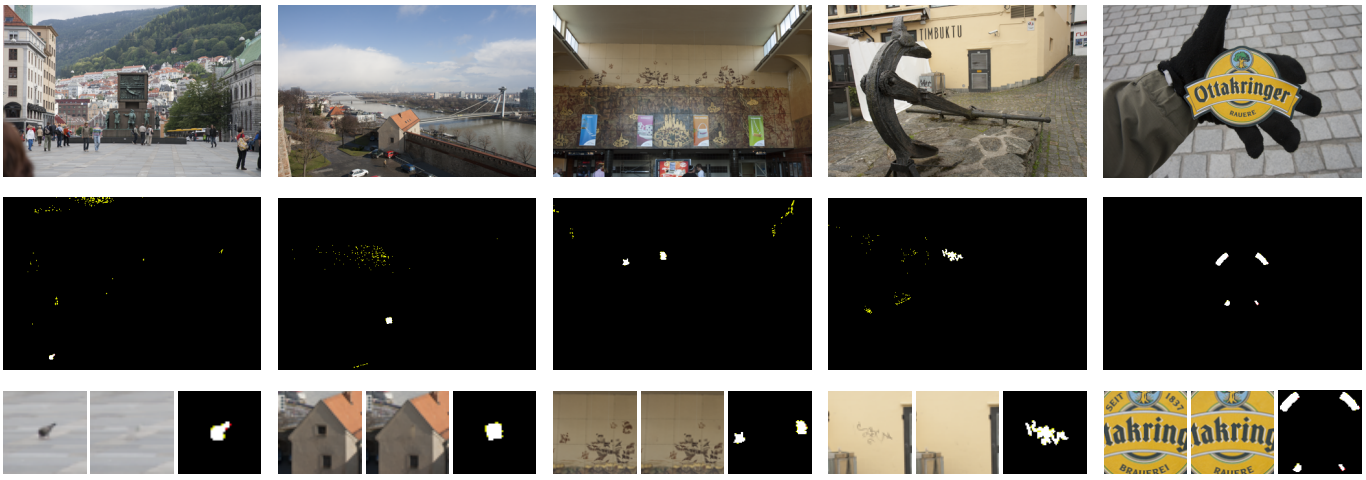


Fig. 9. Examples from the BOSS database. Row 1: inpainted images. Row 2: localization results, where the pixels in white, black, yellow, and red indicate true positive, true negative, false positive, and false negative, respectively. Row 3: the enlarged parts of the ordinal regions, the inpainted regions, and the localization results. The F1-scores for these examples are 0.2713, 0.4690, 0.6801, 0.7339, 0.9759, respectively.

introduced by the imaging pipeline, leading to slightly more false alarms. This phenomenon can be figured out in the results shown in the second row of Fig. 9, where there are several false-alarmed regions outside the inpainted regions. As for the localization performance, the average F1-scores for UCID and BossBase are 0.6562 and 0.5380, respectively. Referring to the examples shown in Fig. 8 and Fig. 9, the cases with F1-scores larger than 0.6 indicate that the inpainted regions can be satisfactorily identified. Although there are some small false-alarmed regions, they do not prevent the identification of the inpainted regions. In the cases with low F1-scores (for example, lower than 0.3), the false-alarmed regions seem to be larger, although the inpainted regions can still be detected. On the whole, the experimental results show that the proposed method can effectively detect and locate the regions altered with diffusion-based inpainting in realistic cases.

C. Computation time

In this subsection, we report the computation time of the proposed method for images of different sizes. On a server equipped with Intel Xeon E5-2690 CPU, 64 GB RAM and Matlab 2015b, we tested five sets of images with size of 256×256, 384×512, 512×512, 1024×1024, 2048×2048, respectively. Each image set contained 100 images. We recorded the computation time for feature extraction, classification, and the two post-processing operations, and show the average computation time in Fig. 10. From this figure, it is observed that the computation time is almost proportional to the total number of pixels within an image, except for morphological filtering on images of 256×256. Among the four processes, feature extraction and classification take the dominant part of computation time. For example, the feature extraction and classification for a 384×512 image (the same size of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2730822, IEEE Transactions on Information Forensics and Security

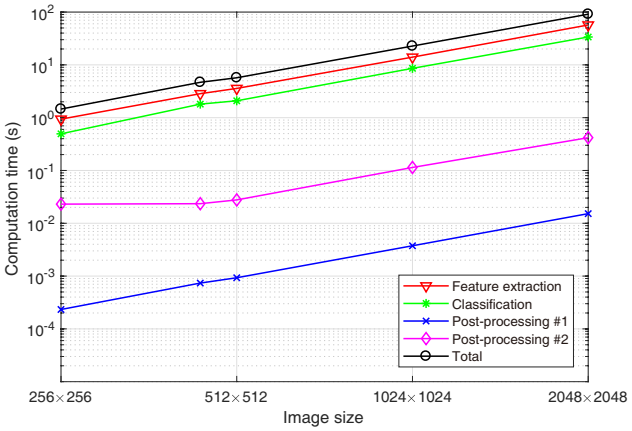


Fig. 10. The computation time for images of different sizes. Post-processing #1 and #2 mean "exclusion of abnormal exposed regions" and "morphological filtering", respectively.

TABLE VI
THE F1-SCORES AND DEGRADATIONS COMPARED TO PLAIN INPAINTING FOR GAMMA CORRECTED IMAGES.

$\gamma = 0.8$			
Algorithm	Isotropic	Edge-oriented	Delaunay-oriented
4096-px	0.8464 (↓0.0488)	0.8296 (↓0.0570)	0.7674 (↓0.0751)
1024-px	0.7167 (↓0.0853)	0.6876 (↓0.1009)	0.5366 (↓0.1316)
256-px	0.5777 (↓0.1042)	0.5517 (↓0.1157)	0.2526 (↓0.1074)
64-px	0.2872 (↓0.0898)	0.2260 (↓0.0843)	0.0609 (↓0.0391)
$\gamma = 1.2$			
Algorithm	Isotropic	Edge-oriented	Delaunay-oriented
4096-px	0.8914 (↓0.0038)	0.8717 (↓0.0149)	0.8152 (↓0.0273)
1024-px	0.7920 (↓0.0099)	0.7625 (↓0.0260)	0.6150 (↓0.0532)
256-px	0.6772 (↓0.0047)	0.6527 (↓0.0147)	0.3398 (↓0.0202)
64-px	0.3780 (↑0.0011)	0.3056 (↓0.0048)	0.0946 (↓0.0054)

the images in UCID database) need about 2.8s and 1.8s, respectively, while the two post-processing operations only take 0.001s and 0.023s. Totally, the computation time for a 384×512 image is about 4.6s.

D. Robustness analysis

In this subsection, we test the robustness of the proposed method to image enhancement (gamma correction), rotation, scaling, and JPEG compression. We respectively applied these post operations to the inpainted images, and performed the localization using the previous model trained on the inpainted images without any post operation. The detailed results and analysis are as follows.

1) Robustness to gamma correction: In this experiment, we applied gamma correction to the inpainted images with parameters γ=0.8 and γ=1.2, respectively, and then tested the proposed method on the gamma corrected inpainted images. The obtained F1-scores for different inpainting algorithms and inpainting sizes are shown in Table VI[4]. From this table, we can observe that the F1-scores for the case of γ=1.2 are nearly the same as those without gamma correction (the maximum degradation is about 0.05), while F1-scores for γ=0.8 encounter sensible degradations. However, the localization results are still satisfactory. On average, the degradations for 4096-px, 1024-px, 256-px, and 64-px inpainted regions are 0.06, 0.10, 0.11, and 0.07, respectively.

[4]Due to page limitation, we do not show the F1-scores for different inpainting shapes independently. For each inpainting algorithm and size, the F1-score reported in the table is the average result for different shapes.

TABLE VII
F1-SCORES AND DEGRADATIONS COMPARED TO PLAIN INPAINTING FOR ROTATED IMAGES.

rotation angle = 5°			
Algorithm	Isotropic	Edge-oriented	Delaunay-oriented
4096-px	0.7275 (↓0.1677)	0.7179 (↓0.1687)	0.6846 (↓0.1579)
1024-px	0.5465 (↓0.2555)	0.5342 (↓0.2543)	0.4537 (↓0.2145)
256-px	0.3771 (↓0.3048)	0.3655 (↓0.3019)	0.2044 (↓0.1556)
64-px	0.1431 (↓0.2339)	0.1093 (↓0.2011)	0.0334 (↓0.0666)
rotation angle = 30°			
Algorithm	Isotropic	Edge-oriented	Delaunay-oriented
4096-px	0.7851 (↓0.1101)	0.7754 (↓0.1111)	0.7238 (↓0.1187)
1024-px	0.6261 (↓0.1759)	0.6088 (↓0.1797)	0.4945 (↓0.1737)
256-px	0.4689 (↓0.2130)	0.4509 (↓0.2165)	0.2290 (↓0.1310)
64-px	0.1495 (↓0.2274)	0.1082 (↓0.2021)	0.0348 (↓0.0652)

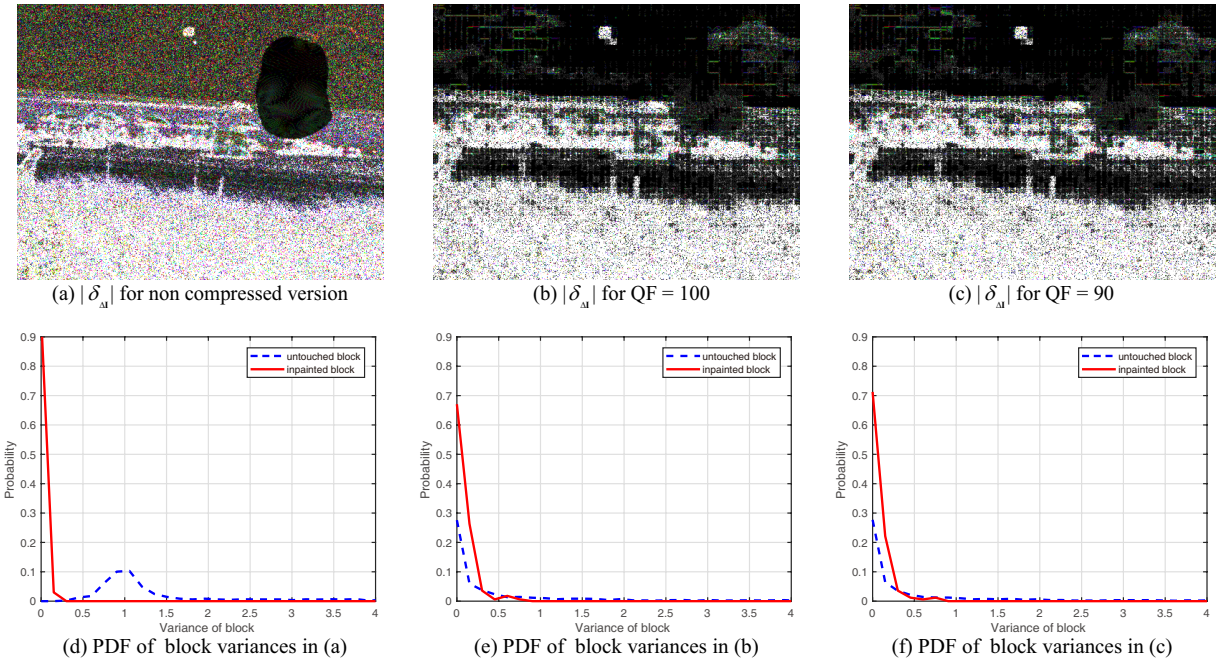
TABLE VIII
THE F1-SCORES AND DEGRADATIONS COMPARED TO PLAIN INPAINTING FOR SCALED IMAGES.

scaling factor = 0.9			
Algorithm	Isotropic	Edge-oriented	Delaunay-oriented
4096-px	0.8322 (↓0.0630)	0.8211 (↓0.0654)	0.7758 (↓0.0667)
1024-px	0.6879 (↓0.1141)	0.6676 (↓0.1209)	0.5503 (↓0.1179)
256-px	0.5292 (↓0.1528)	0.5079 (↓0.1594)	0.2580 (↓0.1020)
64-px	0.1431 (↓0.2338)	0.1135 (↓0.1968)	0.0407 (↓0.0593)
scaling factor = 1.1			
Algorithm	Isotropic	Edge-oriented	Delaunay-oriented
4096-px	0.6829 (↓0.2123)	0.6764 (↓0.2101)	0.6497 (↓0.1928)
1024-px	0.4770 (↓0.3250)	0.4693 (↓0.3192)	0.4105 (↓0.2577)
256-px	0.2960 (↓0.3860)	0.2880 (↓0.3794)	0.1709 (↓0.1891)
64-px	0.1230 (↓0.2540)	0.0969 (↓0.2134)	0.0275 (↓0.0725)

2) Robustness to rotation: In this experiment, we respectively rotated the inpainted image with two rotation angles (i.e., 5° and 30°) using bilinear interpolation, and then applied the proposed method to the two sets of rotated images and obtained the F1-scores as shown in Table VII. Comparing to the results for detecting plain inpainting, the average degradation of F1-scores for rotation with 5° and 30° are 0.21 and 0.16, respectively. When the inpainting regions are of 64-px, the obtained F1-scores are lower than 0.2, meaning that the localization results are not reliable enough in these cases. For the isotropic and edge-oriented inpainting, the F1-scores are very close to or larger than 0.4 when the inpainting sizes are larger than 1024-px, meaning that in such cases the inpainted regions can also be detected with our method.

3) Robustness to scaling: In this experiment, the investigated inpainted images were scaled with bilinear interpolation algorithm. Two scaling factors were tested, i.e., 0.9 and 1.1. The F1-scores obtained on the scaled inpainted images are

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2730822, IEEE Transactions on Information Forensics and Security



(a) $|\delta_{\Delta I}|$ for non compressed version (b) $|\delta_{\Delta I}|$ for QF = 100 (c) $|\delta_{\Delta I}|$ for QF = 90

(d) PDF of block variances in (a) (e) PDF of block variances in (b) (f) PDF of block variances in (c)

Fig. 11. The changes in image Laplacians along the isophote directions for the non-compressed image in Fig. 1 (b) and its JPEG compressed version.

TABLE IX
THE F1-SCORES AND DEGRADATIONS COMPARED TO PLAIN INPAINTING FOR JPEG COMPRESSED IMAGES.

QF = 100			
Algorithm	Isotropic	Edge-oriented	Delaunay-oriented
4096-px	0.4328 (↓0.4624)	0.4103 (↓0.4763)	0.2882 (↓0.5543)
1024-px	0.3149 (↓0.4871)	0.2812 (↓0.5073)	0.1698 (↓0.4985)
256-px	0.2329 (↓0.4491)	0.2016 (↓0.4657)	0.0774 (↓0.2826)
64-px	0.1129 (↓0.2641)	0.0616 (↓0.2487)	0.0098 (↓0.0902)
QF = 90			
Algorithm	Isotropic	Edge-oriented	Delaunay-oriented
4096-px	0.2172 (↓0.6780)	0.1713 (↓0.7153)	0.1472 (↓0.6953)
1024-px	0.0855 (↓0.7165)	0.0631 (↓0.7254)	0.0427 (↓0.6255)
256-px	0.0230 (↓0.6589)	0.0206 (↓0.6468)	0.0093 (↓0.3507)
64-px	0.0036 (↓0.3734)	0.0030 (↓0.3074)	0.0013 (↓0.0987)

shown in Table VIII. From Table VIII, we can observe that up-scaling results in poorer performance than down-scaling, since up-scaling would introduce more blurring artifacts, leading to more false alarms. For down-scaling with the factor 0.9, the propose method can achieve F1-scores larger than 0.5 when the inpainted size is larger than 256-px (except for the Delaunay-oriented inpainting), meaning that the localization performance is still satisfactory. For up-scaling with the factor 1.1, the proposed method can reliably identify the inpainted regions when their sizes are larger than 1024-px.

4) Robustness to JPEG compression: In this experiment, we first compressed the inpainted images to JPEG images with two quality factors (QF) 100 and 90, respectively. Then we applied the proposed method to the JPEG compressed inpainted images and obtained the F1-scores as shown in Table IX. It is observed that JPEG compression would significantly degrade the localization performance. When the size of inpainted region is 4096-px, the average F1-score for QF=100 is 0.3771, while the average F1-score for QF=90 is only 0.1786. For QF=100, the proposed method can still obtain the average F1-score of 0.1706 when the inpainting size is 256-px. However, for QF=90, the proposed method nearly fails to locate the inpainted regions when the inpainting size is smaller than 4096-px. The main reason for such poor performance is that the tampered regions are relatively small (i.e., insufficient statistics), and the JPEG compression would suppress the high-frequency component. Therefore, when computing the changes of image Laplacians along the isophote directions, some untouched smooth regions will present small values and low local variances, which would be confused with the inpainted regions. In Fig. 11, we show the changes in image Laplacians along the isophote directions for the non-compressed image in Fig. 1 (b) and its JPEG compressed version. From subfigures (a), (b) and (c), we observe that the changes of image Laplacians along the isophote directions in the sky and sand beach areas are very similar to those in the inpainted regions after JPEG compression. By comparing the empirical probability density function of block variances shown in subfigures (d)-(f), we observe that the distributions of variances in inpainted and untouched blocks are quite different before JPEG compression, but they become more similar after JPEG compression. Therefore, it will result in many false alarms for the JPEG compressed images and lead to poor localization performance.

We show some example localization maps in Fig. 12. Compared with the results obtained on plain inpainted images, the proposed method can still identify the inpainted regions when the images were undergone gamma correction / rotation / scaling, although producing more false alarms in some cases. For JPEG compressed inpainted images, the localization

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2017.2730822, IEEE Transactions on Information Forensics and Security

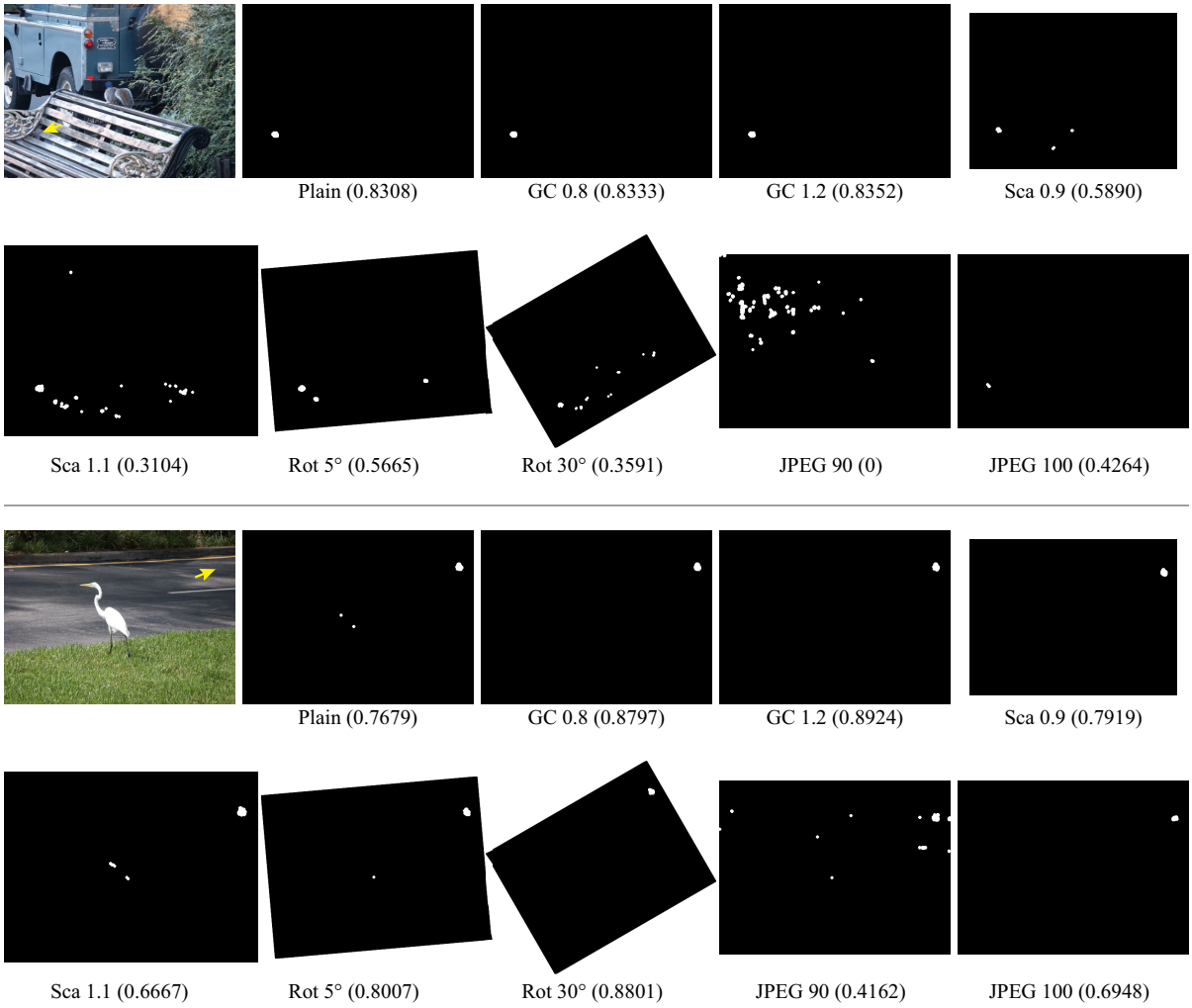


Fig. 12. Example localization maps for different scenarios. The values in parentheses are the obtained F1-scores. The arrows point to the inpainted regions.

results are barely satisfactory when QF=100; when QF=90, too many false alarms appear and thus the performance is poor. In many existing forensic techniques, such as the methods for detecting exemplar-based inpainting, image splicing and camera identification, the detection/localization performance would usually drop a lot after some post operations, especially when the tampered region is small. The authors believe that this work is still important since this is the first step to analyze the tampering artifacts introduced by the diffusion-based inpainting. In the next step, we try to develop some methods to improve the localization performance after some post operations, especially for JPEG compression. For example, combining the proposed method with some JPEG forgery localization methods.

V. Conclusion

In this paper, we have proposed a novel method for the localization of diffusion-based inpainting in digital images. This is the first report devoted to this forensic problem. The proposed method employs some natural artifacts left by the diffusion-based inpainting and designs discriminative features for identifying the inpainted region as well as effective post-processing for refining the localization result. The main contributions of this paper are as follows:

- We have analyzed the diffusion process used in diffusion-based inpainting methods and found that the changes of image Laplacians along the isophote directions are different in the inpainted regions and the untouched regions. This property is the key for detecting diffusion-based inpainting.
- We have adopted the intra-channel and inter-channel local variances as the discriminative features for distinguishing between the inpainted pixels and the untouched pixels. Thus, we can generate a localization map that basically shows the inpainted regions.
- We have introduced the exclusion of the abnormal exposed regions from the localization map, and the refining of the result using morphological filtering with adaptive sizes of the structuring elements. In this way, the localization performance can be further improved.

We have evaluated the proposed method for synthetic inpainted images as well as some realistic examples. The experimental results show that the proposed method is effective for detecting diffusion-based inpainting.

In the future, we will further improve the performance via combining with certain other techniques, such as image segmentation and computer vision. Besides, since the localization performance would be degraded by some post operations, we will enhance the robustness of the proposed method, especially for the robustness to JPEG compression.

REFERENCES

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

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

[3] J. Lukáš, J. Fridrich, and M. Goljan, "Detecting digital image forgeries using sensor pattern noise," in *Proc. SPIE: Security, Steganography, and Watermarking of Multimedia Contents VIII*, vol. 6072, 2006, p. 60720Y.

[4] A. Swaminathan, M. Wu, and K. Liu, "Nonintrusive component forensics of visual sensors using output images," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 1, pp. 91–106, Mar. 2007.

[5] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva, "Image forgery localization via fine-grained analysis of CFA artifacts," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 5, pp. 1566–1577, 2012.

[6] W. Luo, Y. Wang, and J. Huang, "Detection of quantization artifacts and its applications to transform encoder identification," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 4, pp. 810–815, Dec. 2010.

[7] T. Bianchi and A. Piva, "Detection of nonaligned double JPEG compression based on integer periodicity maps," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 842–848, Apr. 2012.

[8] M. Stamm and K. Liu, "Blind forensics of contrast enhancement in digital images," in *Proc. 15th IEEE Int. Conf. Image Process.*, 2008, pp. 3112–3115.

[9] G. Cao, Y. Zhao, R. Ni, and X. Li, "Contrast enhancement-based forensics in digital images," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 3, pp. 515–525, Mar. 2014.

[10] A. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 758–767, 2005.

[11] B. Mahdian and S. Saic, "Blind authentication using periodic properties of interpolation," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 3, pp. 529–538, 2008.

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

[13] C. Chen, J. Ni, and J. Huang, "Blind detection of median filtering in digital images: A difference domain based approach," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4699–4710, Dec. 2013.

[14] D. Cozzolino, G. Poggi, and L. Verdoliva, "Splicebuster: a new blind image splicing detector," in *Proc. IEEE Int. Workshop Inf. Forensics and Security*, 2015, pp. 1–6.

[15] K. Bahrami and A. C. Kot, "Image splicing localization based on blur type inconsistency," in *Proc. IEEE Int. Symposium Circuits and Systems*, 2015, pp. 1042–1045.

[16] E. Ardizzone, A. Bruno, and G. Mazzola, "Copymove forgery detection by matching triangles of keypoints," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 10, pp. 2084–2094, 2015.

[17] J. Li, X. Li, B. Yang, and X. Sun, "Segmentation-based image copy-move forgery detection scheme," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 3, pp. 507–518, 2015.

[18] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. 27th Annual Conf. Computer Graphics and Interactive Techniques*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000, pp. 417–424.

[19] M. M. Oliveira, B. Bowen, R. McKenna, and Y.-S. Chang, "Fast digital image inpainting," in *Proc. Int. Conf. Visualization, Imaging and Image Process.*, Marbella, Spain, Sep. 2001.

[20] T. F. Chan and J. Shen, "Nontexture inpainting by curvature-driven diffusions," *Journal of Visual Communication and Image Representation*, vol. 12, no. 4, pp. 436–449, 2001.

[21] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, Sep. 2004.

[22] T. Ružić and A. Pižurica, "Context-aware patch-based image inpainting using markov random field modeling," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 444–456, Jan. 2015.

[23] X. H. Li, Y. Q. Zhao, M. Liao, F. Y. Shih, and Y. Q. Shi, "Detection of tampered region for JPEG images by using mode-based first digit features," *EURASIP Journal Advances in Signal Process.*, vol. 2012, no. 1, pp. 1–10, 2012.

[24] Y. Q. Zhao, M. Liao, F. Y. Shih, and Y. Q. Shi, "Tampered region detection of inpainting JPEG images," *Optik-International Journal for Light and Electron Optics*, vol. 124, no. 16, pp. 2487–2492, 2013.

[25] I.-C. Chang, J. C. Yu, and C.-C. Chang, "A forgery detection algorithm for exemplar-based inpainting images using multi-region relation," *Image and Vision Computing*, vol. 31, no. 1, pp. 57–71, 2013.

[26] Z. Liang, G. Yang, X. Ding, and L. Li, "An efficient forgery detection algorithm for object removal by exemplar-based image inpainting," *Journal of Visual Communication and Image Representation*, vol. 30, pp. 75–85, 2015.

[27] D. T. Trung, A. Beghdadi, and M.-C. Larabi, "Blind inpainting forgery detection," in *Proc. IEEE Global Conf. Signal and Inf. Process.*, 2014, pp. 1019–1023.

[28] Q. Wu, S.-J. Sun, W. Zhu, G.-H. Li, and D. Tu, "Detection of digital doctoring in exemplar-based inpainted images," in *Proc. Int. Conf. Machine Learning and Cybernetics*, vol. 3, 2008, pp. 1222–1226.

[29] K. S. Bacchuwar, Aakashdeep, and K. R. Ramakrishnan, "A jump patch-block match algorithm for multiple forgery detection," in *Proc. Int. Multi-Conf. on Automation, Computing, Communication, Control and Compressed Sensing*, Mar. 2013, pp. 723–728.

[30] C. Guillemot and O. L. Meur, "Image inpainting : Overview and recent advances," *IEEE Signal Process. Magazine*, vol. 31, no. 1, pp. 127–144, Jan. 2014.

[31] G'MIC. GREYC's Magic for Image Computing. Available at http://gmic.eu.

[32] G. Schaefer and M. Stich, "UCID: an uncompressed color image database," in *Proc. SPIE: Storage and Retrieval Methods and Applications for Multimedia*, vol. 5307, no. 1, 2004, pp. 472–480.

[33] J. Kodovský, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 432–444, 2012.

[34] P. Bas, T. Filler, and T. Pevný, "Break our steganographic system: The ins and outs of organizing BOSS," in *Information Hiding*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, vol. 6958, pp. 59–70, raw images available at http://agents.fel.cvut.cz/stegodata/RAWs.

Haodong Li received the B.S. and Ph.D. degrees from Sun Yat-sen University, Guangzhou, China, in 2012 and 2017, respectively. His research interests include multimedia forensics and security.

Weiqi Luo (M'09-SM'16) received the Ph.D. degree from Sun Yat-sen University, Guangzhou, China, in 2008. He is currently an Associate Professor with the School of Data and Computer, Sun Yat-sen University, and is a Researcher with the Guangdong Key Laboratory of Information Security Technology, Guangzhou. His current research interests include digital multimedia forensics, steganography, and steganalysis.

Jiwu Huang (M'98-SM'00-F'16) received the B.S. degree from Xidian University, Xi'an, China, in 1982, the M.S. degree from Tsinghua University, Beijing, China, in 1987, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 1998. He was with the School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China. He is currently a Professor with the College of Information Engineering, Shenzhen University, Shenzhen, China. His current research interests include multimedia forensics and security. He is also a member of the IEEE Circuits and Systems Society Multimedia Systems and Applications Technical Committee and the IEEE Signal Processing Society Information Forensics and Security Technical Committee. He was a General Co-Chair of the IEEE Workshop on Information Forensics and Security in 2013. He served as an Associate Editor of the IEEE Transactions on Information Forensics and Security from 2010 to 2014. He is a fellow of IEEE.