This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Robust Image Forgery Detection over Online Social Network Shared Images

Haiwei Wu, Jiantao Zhou, Jinyu Tian, and Jun Liu
State Key Laboratory of Internet of Things for Smart City
Department of Computer and Information Science, University of Macau
{yc07912, jtzhou, yb77405, yc07453}@um.edu.mo

Abstract

The increasing abuse of image editing softwares, such as Photoshop and Meitu, causes the authenticity of digital images questionable. Meanwhile, the widespread availability of online social networks (OSNs) makes them the dominant channels for transmitting forged images to report fake news, propagate rumors, etc. Unfortunately, various lossy operations adopted by OSNs, e.g., compression and resizing, impose great challenges for implementing the robust image forgery detection. To fight against the OSN-shared forgeries, in this work, a novel robust training scheme is proposed. We first conduct a thorough analysis of the noise introduced by OSNs, and decouple it into two parts, i.e., predictable noise and unseen noise, which are modelled separately. The former simulates the noise introduced by the disclosed (known) operations of OSNs, while the latter is designed to not only complete the previous one, but also take into account the defects of the detector itself. We then incorporate the modelled noise into a robust training framework, significantly improving the robustness of the image forgery detector. Extensive experimental results are presented to validate the superiority of the proposed scheme compared with several state-of-the-art competitors. Finally, to promote the future development of the image forgery detection, we build a public forgeries dataset based on four existing datasets and three most popular OSNs. The designed detector recently won the top ranking in a certificate forgery detection competition[1]. The source code and dataset are available at ***https://github.com/HighwayWu/ImageForensicsOSN***.

1. Introduction

The ever-increasing popularity of powerful image editing softwares has made the manipulation of images an extremely easy task. The manipulated or forged images are becoming increasingly dangerous in various fields such as removing copyright watermarks, producing fake news, and being forged evidence in court. Meanwhile, with the vigorous development of the Internet, online social networks (OSNs) have become dominant platforms for information transmission, where images occupy a large portion. Many forged images are transmitted over various OSNs, influencing people's opinion towards *e.g.*, important documents (certificates), commercial products, political issues, etc.

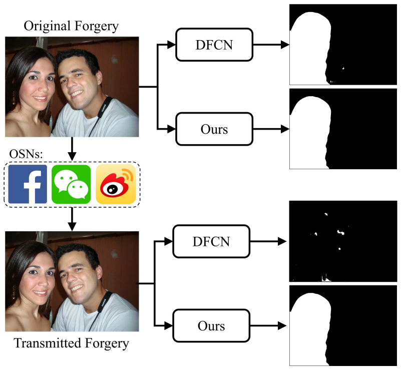

Figure 1. The detection results of DFCN [42] and ours by using an original forgery and the one transmitted through an OSN. The left girl is spliced (forged).

A large number of methods [4, 5, 8, 11, 12, 18, 20–23, 26, 27, 36, 37, 39, 42] have been proposed to detect and localize image forgery. Some of them are designed to detect specific forms of tampering, such as splicing [18, 26], copy-move [23, 39] and inpainting [21, 22, 36], while the others aim to identify more complex or compound forgeries. However, few research has been done to explicitly address the design of the robust forgery detection against the lossy operations in the ubiquitous OSN platforms. Such a topic is very important because these lossy operations can severely degrade the detection performance. As shown in Fig. 1, the state-of-the-art algorithm [42] can accurately detect the forged regions from the original forgery; but the detection performance would be severely degraded when handling the forgery transmitted through Facebook.

For mitigating the negative impacts of OSNs, the first

[1] https://tianchi.aliyun.com/competition/entrance/531812/introduction

critical issue is to analyze and model the noise introduced by the OSN lossy channels. However, this is a rather difficult problem mainly because current platforms do not disclose the process for manipulating the transmitted images. Although some works [33, 34] revealed part of the operations adopted by OSNs, there are still many unknown ones, *e.g.*, the enhancement filtering in Facebook. More importantly, OSNs often adjust their image processing pipelines, making the modeling even more challenging.

To deal with the aforementioned challenges, in this paper, we aim to design a robust image forgery detection method to defeat the lossy operations in OSNs. Specifically, for dealing with the OSN degradations, we propose a noise modeling scheme and integrate the mimetic noises into a robust training framework. We decouple the OSN noises into two components: 1) *predictable noise* and 2) *unseen noise*. The former is designed to simulate the predictable loss brought by known operations, (*e.g.*, JPEG compression), whose modeling relies on a deep neural network (DNN) with the residual learning and an embedded differentiable JPEG layer. While the latter is mainly in response to the unknowable actions conducted by OSNs and/or the discrepancy between the training and testing of various OSNs. Apparently, it is unrealistic to build a suitable model for the unseen noise from the perspective of the signal itself. To address this difficulty, we transfer our observations from the noise perspective to the forgery detector, only focusing on the noise that may cause deterioration of the detection performance. Such a strategy naturally incubates a new algorithm to model the unseen noise by utilizing the core idea of *adversarial noise* [35], which is essentially imperceptible perturbation that can severely degrade the model performance. It is shown that our robust image forgery detection method demonstrates superior robustness and outperforms several state-of-the-art algorithms. An example of the detection result of our scheme is shown in Fig. 1. Finally, we build a public forgeries dataset with more than 5000 items based on four existing datasets [1, 6, 14, 17] and three OSN platforms (Facebook, Weibo, and Wechat). Our major contributions can be summarized as follows:

- We propose a novel training scheme for robust image forgery detection against transmission over OSNs. The training scheme not only models the predictable noise introduced by OSNs, but also incorporates the unseen noise to further promote the robustness.
- Our model achieves better detection performance in comparison with several state-of-the-art methods [12, 27, 37, 42], especially in the scenario of fighting against the transmission over OSNs.
- We build a public forgeries dataset based on four existing datasets [1, 6, 14, 17] and three platforms (Facebook, Weibo, and Wechat).

The rest of this paper is organized as follows. Sec. 2 reviews the related works. Sec. 3 presents the details on the robust image forgery detection through the proposed robust training strategy. Experimental results are given in Sec. 4 and Sec. 5 concludes.

2. Related works

2.1. Image forgery detection

Many forensics methods (*e.g.*, [2, 3, 5, 7, 8, 18–23, 26, 36, 39, 40]) have been proposed to verify the authenticity of digital images. These methods detect the forged regions through *specific* artifacts left by, *e.g.*, splicing [18, 26], copy-move [23, 39], median filtering [8, 20], inpainting [21, 22, 36], etc. To better fit the practical requirements, more and more methods have been developed to address the problem of detecting general (compound) types of forgeries [4, 5, 11, 12, 27, 37, 41, 42], among which deep learning based methods are the most successful. Along this line, Wu *et al*. [37] proposed the MT-Net, a general forgery detection network, which first extracts image manipulation features and then identifies anomalous regions. Mayer and Stamm recently [27] introduced the forensic similarity to determine whether two image patches contain the same forensic traces. From the perspective of the camera fingerprint, Cozzolino and Verdoliva [12] designed a method for extracting a camera model fingerprint, called noiseprint, so as to disclose the forged regions. For learning the traces of generic forgeries, Zhuang *et al*. [42] utilized a training data generation strategy by using the Photoshop scripting.

2.2. Online Social Network (OSN)

The popularity of various OSN platforms, *e.g.*, Facebook, Wechat, Weibo, etc, significantly simplifies the dissemination and sharing of images. However, as indicated by many existing works [33, 34], almost all OSNs manipulate the uploaded images in a lossy fashion. The noise introduced by these lossy operations could severely affect the effectiveness of forensic methods. Taking Facebook as an example, as discovered in [32–34], these manipulations mainly consist of three stages: resizing, enhancement filtering, and JPEG compression. Specifically, resizing would be applied if the resolution of the image is above 2048 pixels. After that, some selected blocks in the image undergo highly adaptive and complex enhancement filtering. As mentioned in [33, 34], it is very challenging to precisely know these enhancement filtering operations due to their adaptiveness. Finally, the image is subject to a round of JPEG compression with a quality factor (QF) *adaptively* determined according to the image content. Through the analysis of the dataset provided in [33], the QF values used by Facebook range from 71 to 95. Although the image manipulations on different OSN platforms are different, the operations conducted by mainstream OSNs still share many similarities (*e.g.*, ubiquitous JPEG compression) [33].

Some existing forensics [9, 24, 38] are designed to identify the involved transmission operations. Liao *et al.* [9, 24] first raised a feature decoupling method for the identification of two manipulations based on blind signal separation. To further reveal a long chain, You *et al.* [38] presented a solution by innovatively representing the manipulation chain detection as a machine translation problem.

3. Robust image forgery detection against transmission over OSNs

In this section, we present the details on the robust image forgery detection scheme against the transmission over OSNs. The key technique leading to the success is to appropriately model the degradations incurred by OSNs, and integrate such knowledge into a robust training framework. Recall from Sec. 2.2 that the image processing operations in an OSN are rather complicated; some of them can be precisely known, while some others can only be partially known or even completely unknown. Therefore, we propose to divide the OSN noise into two types: 1) predictable noise and 2) unseen noise. The former type corresponds to the case that the degradation source is clearly identified. While the latter type is a combination of various noise uncertainties, including the unknown modeling/parameters, the discrepancy between the training and testing OSNs, and even some totally unseen degradation sources. By adding the modelled OSN noise in the training phase, the detector is expected to learn more generalized features that could survive the OSN transmission, making the overall forgery detection performance significantly improved.

In Fig. 2, we illustrate the framework of our robust training scheme for the forgery detection, which consists of four stages. Roughly speaking, Stage 1 and Stage 2 are devoted to simulate the predictable noise via a differentiable network. Stage 3 deals with the modeling of the unseen noise through an adversarial noise generation strategy. Eventually, Stage 4 handles the actual robust training of the image forgery detector $f_{\boldsymbol{\theta}}$. Note that our robust training scheme can be incorporated with any deep learning based image forgery detectors. As the focus of this work is more on the robust training, we in the following restrict our attention to the Stages 1-3, while leaving the details of $f_{\boldsymbol{\theta}}$ in Sec. 4.1.

Formally, let $\boldsymbol{\tau}$ and $\boldsymbol{\xi}$ denote the predictable noise and unseen noise, respectively, and hence the compound noise considered in the robust training stage becomes

$$\boldsymbol{\delta} = \boldsymbol{\tau} + \boldsymbol{\xi}. \tag{1}$$

For each training iteration, we first sample two pristine 3-channel (RGB) color images $\{\mathbf{p}_1, \mathbf{p}_2\} \in \mathbb{R}^{H\times W\times 3}$, and one binary mask $\mathbf{y} \in \{0,1\}^{H\times W\times 1}$, where 1's are assigned to the forged regions and 0's elsewhere. Then a forged image $\mathbf{x}$ can be synthesized as

$$\mathbf{x} = \mathbf{p}_1 \odot (1-\mathbf{y}) + \mathbf{p}_2 \odot \mathbf{y}, \tag{2}$$

where $\odot$ denotes the element-wise multiplication. Upon having the pairs of forged image and the corresponding ground-truth mask, we can create a dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$ for the training, where i is the index for the training sample. The robust training of the image forgery detector $f_{\boldsymbol{\theta}}$ under the compound noise $\boldsymbol{\delta}$ can then be formulated as:

$$\arg\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{P(\boldsymbol{\delta})}\Big\{\mathcal{L}_b(f_{\boldsymbol{\theta}}(\mathbf{x}_i + \boldsymbol{\delta}), \mathbf{y}_i)\Big\}, \tag{3}$$

where $P(\boldsymbol{\delta})$ denotes the distribution of the compound noise $\boldsymbol{\delta}$, N is the number of training samples, and $\mathcal{L}_b$ is the binary cross-entropy (BCE) loss.

In our noise model, we consider a rather general setting that the two noise components $\boldsymbol{\tau}$ and $\boldsymbol{\xi}$ are dependent. Then, the robust training scheme Eq. (3) can be further written as

$$\arg\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{P(\boldsymbol{\tau})}\Big\{\mathbb{E}_{P(\boldsymbol{\xi}|\boldsymbol{\tau})}\{\mathcal{L}_b(f_{\boldsymbol{\theta}}(\mathbf{x}_i + \boldsymbol{\tau} + \boldsymbol{\xi}), \mathbf{y}_i)\}\Big\}, \tag{4}$$

where $P(\boldsymbol{\tau})$ is the marginal distribution of $\boldsymbol{\tau}$, and $P(\boldsymbol{\xi}|\boldsymbol{\tau})$ is the conditional distribution of $\boldsymbol{\xi}$ given $\boldsymbol{\tau}$. From the implementation perspective, such expected values could be efficiently and accurately computed upon having a sufficient number of noise samples. To conduct the robust training given in Eq. (4), a crucial task is to model the marginal distribution $P(\boldsymbol{\tau})$ and the conditional distribution $P(\boldsymbol{\xi}|\boldsymbol{\tau})$.

3.1. Modeling the distribution $P(\boldsymbol{\tau})$

We now model the distribution $P(\boldsymbol{\tau})$, where the degradation is caused by the lossy operations of OSN platforms. From Sec. 2.2, we know that the dominating degradation source of $\boldsymbol{\tau}$ is the applied JPEG compression, and the post-processing (*e.g.*, enhancement filtering) also partially contributes to $\boldsymbol{\tau}$. For an image $\mathbf{x}_i$ and a *fixed* OSN platform, the incurred noise can be easily calculated by

$$\boldsymbol{\tau}_i = \mathrm{OSN}(\mathbf{x}_i) - \mathbf{x}_i, \tag{5}$$

where the function $\mathrm{OSN}(\cdot)$ reflects all the operations conducted by the given OSN platform. Note that $\boldsymbol{\tau}_i$ depends on $\mathbf{x}_i$, namely, the noise is signal dependent. Seemingly, in this way, we can generate a lot of noise samples, which can be used to model the distribution of $P(\boldsymbol{\tau})$. However, in practice, such a naive modeling scheme is quite problematic. The processed image $\mathrm{OSN}(\mathbf{x}_i)$ has to be obtained by uploading $\mathbf{x}_i$ to the specific OSN platform, and then downloading it. Such procedure, on one hand, is time-consuming; on the other hand, many OSNs do not allow

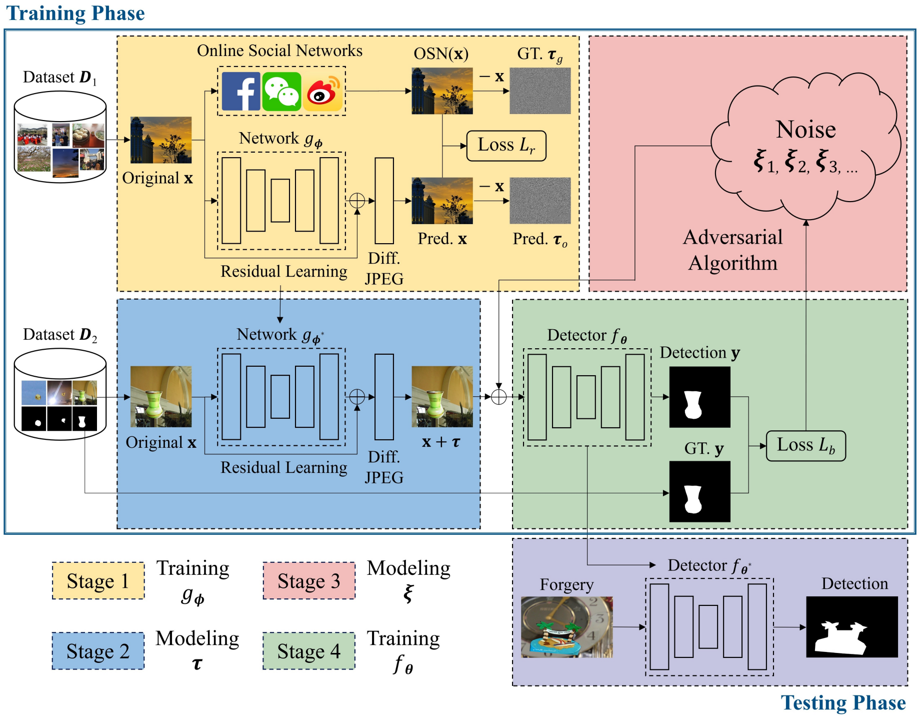

Figure 2. The overview of our proposed training scheme and the corresponding testing phase.

too many times of the uploading/downloading operations. For instance, Weibo even bans the account if too many uploading operations are observed in a short period of time. This seriously limits the number of obtained noise samples, making such a naive scheme highly ineffective in practice.

To resolve this challenge, we resort to another strategy of modeling $P(\boldsymbol{\tau})$ in an inexplicit manner. We propose to use a substitute deep network for mimicking the OSN operations, so as to conveniently produce a large number of noise samples $\boldsymbol{\tau}_i$. Specifically, to be consistent with the image processing pipeline in the OSN platform, we train a DNN model, which explicitly embeds a differentiable layer to describe the JPEG compression. For an input image $\mathbf{x}_i$, we aim to learn a mapping $g_{\boldsymbol{\phi}} : \mathbb{R}^d \to \mathbb{R}^d$, where $g_{\boldsymbol{\phi}}$ is a network with trainable parameters $\boldsymbol{\phi}$, predicting the OSN output. We employ a U-Net architecture [28] for $g_{\boldsymbol{\phi}}$, as it is essentially an image-to-image mapping. The training procedure is illustrated in Stage 1 of Fig. 2, and then the well-trained $g_{\boldsymbol{\phi}^*}$ is employed in the Stage 2 for modeling $P(\boldsymbol{\tau})$. At the training stage, we collect pairs of input image $\mathbf{x}_i \in \mathbb{R}^d$ and the OSN transmitted version $\mathrm{OSN}(\mathbf{x}_i) \in \mathbb{R}^d$ in an offline manner. The objective function for training $g_{\boldsymbol{\phi}}$ can be formulated as

$$\min_{\boldsymbol{\phi}} \Big\{ \mathcal{L}_r(g_{\boldsymbol{\phi}}(\mathbf{x}_i), \mathrm{OSN}(\mathbf{x}_i)) \Big\}, \tag{6}$$

where $\mathcal{L}_r(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2$.

As we are more interested in learning the noise incurred by the OSN transmission rather than the image content itself, we adopt a residual learning structure [16] when designing $g_{\boldsymbol{\phi}}$. Bearing this in mind, we change the objective function into

$$\min_{\boldsymbol{\phi}} \Big\{ \mathcal{L}_r(\mathbf{x}_i + g_{\boldsymbol{\phi}}(\mathbf{x}_i), \mathrm{OSN}(\mathbf{x}_i)) \Big\}. \tag{7}$$

The residual learning is beneficial for the model optimization, significantly boosting the modeling performance.

Furthermore, we explicitly integrate a special JPEG layer into the model for better generating the structural, JPEG-like artifacts, which reflects the true situation in various OSN platforms. To enable the end-to-end optimization of the objective function in Eq. (7), we need to ensure that every step of the JPEG compression remains differentiable. It is easy to find that the quantization is the only non-differentiable step, mainly because the employed rounding function $\lfloor \cdot \rceil$ has 0 derivative everywhere. To deal with it, we approximate the rounding function with a differentiable version [31]:

$$\lfloor x \rceil_a = \lfloor x \rceil + (x - \lfloor x \rceil)^3. \tag{8}$$

Upon having a differentiable JPEG layer, the objective function for training $g_{\boldsymbol{\phi}}$ becomes

$$\min_{\boldsymbol{\phi}} \mathcal{L}_r(\mathcal{J}_q(\mathbf{x}_i + g_{\boldsymbol{\phi}}(\mathbf{x}_i)), \mathrm{OSN}(\mathbf{x}_i)), \tag{9}$$

where $\mathcal{J}_q$ represents the differentiable JPEG layer with a given QF q. In our training, q is uniformly sampled in the range $[71, 95]$ as adopted by Facebook. It is then straightforward to derive the noise $\boldsymbol{\tau}_i$ as

$$\boldsymbol{\tau}_i(q) = \mathcal{J}_q(\mathbf{x}_i + g_{\boldsymbol{\phi}^*}(\mathbf{x}_i)) - \mathbf{x}_i, \tag{10}$$

where $\boldsymbol{\phi}^*$ is obtained by solving the optimization problem Eq. (9) and q is the QF associated with the JPEG compression. Monte Carlo (MC) sampling scheme can then be implemented to generate a large number of noise samples for modeling the distribution $P(\boldsymbol{\tau})$.

3.2. Modeling the conditional distribution $P(\boldsymbol{\xi}|\boldsymbol{\tau})$

We then tackle the issue of modeling the conditional distribution $P(\boldsymbol{\xi}|\boldsymbol{\tau})$ so that we can solve the optimization problem in Eq. (4). The reason why we incorporate the noise term $\boldsymbol{\xi}$ is that the predictable noise $\boldsymbol{\tau}$ cannot fully capture the noise behavior encountered in practice. For instance, different OSNs may adopt distinct procedures, *e.g.*, adjusting the QF dynamically, performing resizing adaptively, or even introducing completely unknown operations.

A critical problem now is how to build a proper model for the unseen noise $\boldsymbol{\xi}$. Obviously, it is unrealistic to model $\boldsymbol{\xi}$ from the characteristic of the signal itself, as we do in Sec. 3.1. To resolve this challenge, we shift our position from the noise aspect to the detector $f_{\boldsymbol{\theta}}$, by studying the noise effect on the detection performance. Among the various underlying unseen noise $\boldsymbol{\xi}$, we actually only need to pay attention to the ones that degrade the detection performance, while neglecting those that have little effect to the detection. This motivates us to employ a type of *adversarial noise* [35] when modeling $P(\boldsymbol{\xi}|\boldsymbol{\tau})$. Essentially, adversarial noises are generally imperceptible to the human senses while being able to cause severe model output errors. Meanwhile, the unseen noise $\boldsymbol{\xi}$ that we focus on is the one capable of fooling the detector and is also usually small (a highly distorted image would deviate from the purpose of making a forgery). Such a similarity in terms of the effect to the detector $f_{\boldsymbol{\theta}}$ makes the adversarial noise a suitable candidate for modeling $\boldsymbol{\xi}$.

From the adversarial point of view, there are various ways of defining the noise $\boldsymbol{\xi}$, as long as the adversarial example, created by adding the noise $\boldsymbol{\xi}$ to the original normal example, goes across the decision boundary. Noticing the fact that the noise $\boldsymbol{\xi}$ is typically of small amplitude, we propose to set the direction of $\boldsymbol{\xi}$ along the gradient of the cost function with respect to the input, so as to minimize the noise energy. Therefore, for a given input $\mathbf{x}_i$, the predictable noise $\boldsymbol{\tau}_i$, and the target output $\mathbf{y}_i$, the unseen noise $\boldsymbol{\xi}_i$ is formulated as

$$\boldsymbol{\xi}_i = \mathcal{S}(\nabla_{\mathbf{x}_i}\mathcal{L}_b(f_{\boldsymbol{\theta}}(\mathbf{x}_i + \boldsymbol{\tau}_i), \mathbf{y}_i)), \tag{11}$$

where $\mathcal{S}$ returns the sign of the gradient. By adding such adversarial noises during the training, it is expected to make the learned model robust against not only the specific adversarial noise but also more general unseen noise.

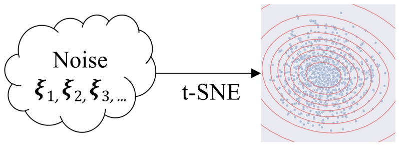
Noise
$\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \boldsymbol{\xi}_3, ...$
t-SNE

Figure 3. Visualization of 1000 $\boldsymbol{\xi}$ samples by using t-SNE [13].

However, the noise calculated by Eq. (11) depends on the specific input $\mathbf{x}_i$, rather than a general one applicable to all the examples in the training set and unknown examples. For comprehensively enhancing the generalization ability of the detector, we propose to adjust the direction of the adversarial noise to a global gradient direction. To this end, we adopt a strategy similar to the Stochastic Gradient Descend (SGD) [30], by a stochastic approximation approach from randomly selected subsets of the training dataset. More specifically, for the $(t+1)$-th input $\mathbf{x}_{t+1}$, the $\boldsymbol{\xi}_{t+1}$ (conditioning on $\boldsymbol{\tau}$) could be set as the average gradient calculated from the first t inputs, namely,

$$\boldsymbol{\xi}_{t+1} = \frac{1}{t}\sum_{i=0}^{t}\mathcal{S}(\nabla_{\mathbf{x}_i}\mathcal{L}_b(f_{\boldsymbol{\theta}}(\mathbf{x}_i + \boldsymbol{\tau}_i + \boldsymbol{\xi}_i), \mathbf{y}_i)), \tag{12}$$

where $\boldsymbol{\xi}_0$ is initialized as $\mathbf{0}$. Although Eq. (12) can be used to estimate the average gradients, it only reflects the gradients of specific known data (the training data). To alleviate the aforementioned problem and further improve the robustness, we propose to perturb the $\boldsymbol{\xi}_{t+1}$ in a small range. Here, it would be more ideal to use a parametric model to characterize the average gradients. To find an appropriate model for the average gradient, we first take a data-driven approach, analyzing the statistics of 1000 samples of $\boldsymbol{\xi}$ that are randomly selected from the training process. In Fig. 3, we visualize these 1000 random samples in a 2D space by using the t-SNE [13]. It can be seen that the sample points are concentrated around a certain center, and gradually vanish when they move away from the center. This phenomenon suggests us to use a Gaussian distribution for modeling the average gradient, *i.e.*,

$$\boldsymbol{\xi}_{t+1}|\boldsymbol{\tau} \sim \mathcal{N}(\boldsymbol{u}_{t+1}, \sigma^2\mathbf{I}), \tag{13}$$

where σ is an empirically set parameter for controlling the variance,

$$\boldsymbol{u}_{t+1} = \epsilon \cdot \frac{1}{t}\sum_{i=0}^{t}\mathcal{S}(\nabla_{\mathbf{x}_i}\mathcal{L}_b(f_{\boldsymbol{\theta}}(\mathbf{x}_i + \boldsymbol{\tau}_i + \boldsymbol{\xi}_i), \mathbf{y}_i)), \tag{14}$$

and ϵ is a parameter used for constraining the magnitude of the perturbations to avoid unnecessary model degradation.

Upon having the parametric model in Eq. (13), we can easily generate noise samples for modeling the conditional

Algorithm 1: The training algorithm

```
Input: Training datasets $\mathcal{D}_1$ and $\mathcal{D}_2$; training epochs $N_1$
       and $N_2$; learning rates $l_{\boldsymbol{\phi}}$ and $l_{\boldsymbol{\theta}}$.
Output: Trained detector $f_{\boldsymbol{\theta}^*}$
Randomly initialize $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$
for epoch = 1 to $N_1$ do
    for minibatch $(\mathbf{x}_i, \mathbf{y}_i) \subset \mathcal{D}_2$ do
        $\mathbf{g}_{\boldsymbol{\phi}} = \nabla_{\boldsymbol{\phi}}[\mathcal{L}_r(\mathcal{J}_q(\mathbf{x}_i + g_{\boldsymbol{\phi}}(\mathbf{x}_i)), \mathbf{y}_i)]$    ▷ Eq. (9)
        $\boldsymbol{\phi} = \boldsymbol{\phi} - l_{\boldsymbol{\phi}} \cdot \mathbf{g}_{\boldsymbol{\phi}}$    ▷ Update $g_{\boldsymbol{\phi}}$
    end
end
Temporary output $g_{\boldsymbol{\phi}^*} = g_{\boldsymbol{\phi}}$
Initialize $\boldsymbol{u}_0 = \mathbf{0}$
for epoch = 1 to $N_2$ do
    for minibatch $(\mathbf{x}_i, \mathbf{y}_i) \subset \mathcal{D}_2$ do
        Initialize $\mathbf{L}_0 = \mathbf{0}$
        for $j$ = 1 to $m$ do
            $q_j \sim \text{Uniform}(71, 95)$    ▷ Sample QF
            $\boldsymbol{\tau}_j = \mathcal{J}_{q_j}(\mathbf{x}_i + g_{\boldsymbol{\phi}^*}(\mathbf{x}_i)) - \mathbf{x}_i$    ▷ Model $\boldsymbol{\tau}$
            $\{\boldsymbol{\xi}_1, \cdots, \boldsymbol{\xi}_h\} \sim \mathcal{N}(\boldsymbol{u}_{i-1}, \sigma^2\mathbf{I})$    ▷ Model $\boldsymbol{\xi}$
            $\mathbf{L}_j = \mathbf{L}_{j-1} + \sum_{k=1}^{h} \mathcal{L}_b(f_{\boldsymbol{\theta}}(\mathbf{x}_i + \boldsymbol{\tau}_j + \boldsymbol{\xi}_k), \mathbf{y}_i)$
                ▷ Eq. (15)
        end
        $\mathbf{g}_{\boldsymbol{\theta}} = \nabla_{\boldsymbol{\theta}}\mathbf{L}_m, \mathbf{g}_{\mathbf{x}_i} = \nabla_{\mathbf{x}_i}\mathbf{L}_m$
        $\boldsymbol{\theta} = \boldsymbol{\theta} - l_{\boldsymbol{\theta}} \cdot \mathbf{g}_{\boldsymbol{\theta}}$    ▷ Update $f_{\boldsymbol{\theta}}$
        $\boldsymbol{u}_i = \boldsymbol{u}_{i-1} + \epsilon \cdot \mathcal{S}(\mathbf{g}_{\mathbf{x}_i})$    ▷ Eq. (14)
    end
end
Final output $f_{\boldsymbol{\theta}^*} = f_{\boldsymbol{\theta}}$
```

distribution $P(\boldsymbol{\xi}|\boldsymbol{\tau})$. Thereupon, Eq. (4) can be expanded as

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \sum_{j=1}^{m} \sum_{k=1}^{h} \mathcal{L}_b(f_{\boldsymbol{\theta}}(\mathbf{x}_i + \boldsymbol{\tau}_j + \boldsymbol{\xi}_k), \mathbf{y}_i), \tag{15}$$

where the expectations with respect to $\boldsymbol{\tau}$ and $\boldsymbol{\xi}$ are approximated with m and h MC samples, respectively. With this computable loss function, we are able to perform the robust training, as summarized in Algorithm 1.

4. Experimental results

In this section, we present the experimental results to show the superior performance of our proposed method. Due to the space limit, more results are given in the supplementary file.

4.1. Experimental setup

Baseline detector. The detector aims to detect the forged regions at the pixel level accuracy. Specifically, the detector $f_{\boldsymbol{\theta}} : \mathbb{R}^{H\times W\times 3} \to \mathbb{R}^{H\times W\times 1}$ takes a color image with the resolution $H \times W$ as input, and eventually outputs the binary map for the detection result. In our implementation, we adopt the U-Net [28] architecture in the baseline detector. To improve the capability of extracting forgery relevant features, we further augment the architecture by incorporating the spatial channel "Squeeze-and-Excitation (SE)" mechanism [29], resulting a variant called *SE-U-Net*, rather than simply using the traditional vanilla U-Net.

Training/Validation datasets. For the training of the OSN network $g_{\boldsymbol{\phi}}$, we adopt the dataset **WEI** (denoted as $\mathcal{D}_1$) [33], which contains over 1300 original images and their processed versions upon the transmission over Facebook. It should be noted that we only use the data from Facebook for training $g_{\boldsymbol{\phi}}$. While for the training of $f_{\boldsymbol{\theta}}$, we use the **Dresden** [15] dataset as the source of pristine images. We then generate the forged images by splicing the pristine images with the objects from the **MS-COCO** [25] dataset. The dataset of these forged images is denoted as $\mathcal{D}_2$. Also, $\mathcal{D}_1$ and $\mathcal{D}_2$ are randomly divided into training and validation sets with the ratio of $9:1$.

Testing datasets. We create testing datasets by adopting four widely-used ones (**DSO** [6], **Columbia** [17], **NIST** [1] and **CASIA** [14]), and producing their OSN-transmitted versions. More specifically, we manually upload and download the aforementioned datasets over three most popular OSNs (Facebook, Wechat and Weibo), resulting in OSN-transmitted datasets with 5232 forgeries and corresponding masks. These collected datasets are made available at **https://github.com/HighwayWu/ImageForensicsOSN**. We hope that these datasets can serve as useful benchmarks to our research community for fighting against the forgeries shared over OSNs.

Competitors. We compare our proposed scheme with four state-of-the-art methods: **MT-Net** [37], **NoiPri** [12], **ForSim** [27], and **DFCN** [42].

4.2. Quantitative comparisons

The quantitative comparisons in terms of the AUC, F1 and IoU (higher are better) in the pixel domain are presented in Tab. 1. Here we also report the results of the baseline detector for demonstrating the improvement of our robust training scheme in a comparative way. As can be observed, when the forgeries are not transmitted through an OSN, the detection methods **ForSim** [27], **DFCN** [42] and ours achieve comparable results, while **MT-Net** [37] and **NoiPri** [12] perform slightly worse. It should be noted that, **NoiPri** cannot be applied to detect the forgeries in **CASIA** due to their small resolutions, while our method has no such limitation and perform even better than the other competitors on **CASIA**.

In the scenario that the forgeries are passed through OSNs, the detection performance of all existing methods has deteriorated significantly. For instance, after the transmission over Facebook, Weibo and Wechat, the IoU scores associated with **MT-Net** drop by 10.1%, 11.1%, and 9.4%,

Models	OSNs	Test Datasets														
		DSO [6]			**Columbia** [17]			**NIST** [1]			**CASIA** [14]			Average		
		AUC	F1	IoU	AUC	F1	IoU	AUC	F1	IoU	AUC	F1	IoU	AUC	F1	IoU
MT-Net [37]	-	.795	.344	.253	.747	.357	.258	.634	.088	.054	.776	.130	.086	.738	.230	.163
NoiPri [12]	-	**.902**	.339	.253	.840	.362	.260	.672	.119	.078	-	-	-	.804	.273	.197
ForSim [27]	-	.796	**.487**	**.371**	.731	.604	.474	.642	.188	.123	.554	.169	.102	.681	.362	.268
DFCN [42]	-	.724	.303	.227	.789	.541	.395	.778	.250	.204	.654	.192	.119	.736	.322	.236
Baseline	-	.761	.312	.194	.763	.616	.501	.682	.221	.139	.774	.402	.342	.745	.388	.294
Ours	-	.854	.436	.308	**.862**	**.707**	**.608**	**.783**	**.332**	**.255**	**.873**	**.509**	**.465**	**.843**	**.496**	**.409**
MT-Net [37]	Facebook	.638	.109	.071	.626	.103	.056	.652	.095	.057	.763	.102	.065	.670	.102	.062
NoiPri [12]	Facebook	.777	.150	.097	.722	.223	.143	.583	.057	.034	-	-	-	.694	.143	.091
ForSim [27]	Facebook	.689	.356	.238	.607	.450	.304	.580	.140	.085	.537	.157	.094	.603	.276	.180
DFCN [42]	Facebook	.673	.238	.184	.687	.479	.338	.705	.207	.138	.654	.190	.116	.680	.278	.194
Baseline	Facebook	.714	.180	.105	.689	.594	.497	.646	.200	.136	.728	.350	.298	.694	.331	.259
Ours	Facebook	**.859**	**.447**	**.320**	**.883**	**.714**	**.611**	**.783**	**.329**	**.253**	**.862**	**.462**	**.417**	**.847**	**.488**	**.400**
MT-Net [37]	Weibo	.606	.057	.036	.620	.103	.056	.671	.088	.053	.754	.099	.063	.663	.087	.052
NoiPri [12]	Weibo	.606	.093	.061	.664	.175	.108	.580	.054	.030	-	-	-	.616	.107	.066
ForSim [27]	Weibo	.568	.260	.165	.610	.453	.312	.581	.150	.094	.542	.165	.100	.575	.257	.168
DFCN [42]	Weibo	.639	.227	.140	.676	.458	.319	.706	.192	.125	.653	.191	.117	.668	.267	.175
Baseline	Weibo	.703	.120	.073	.681	.558	.477	.683	.163	.116	.762	.338	.310	.707	.294	.244
Ours	Weibo	**.808**	**.370**	**.253**	**.883**	**.724**	**.626**	**.780**	**.294**	**.219**	**.858**	**.466**	**.421**	**.832**	**.463**	**.380**
MT-Net [37]	Wechat	.582	.076	.045	.613	.199	.125	.654	.095	.057	.724	.080	.048	.643	.113	.069
NoiPri [12]	Wechat	.618	.098	.062	.639	.202	.124	.575	.041	.026	-	-	-	.610	.114	.070
ForSim [27]	Wechat	.564	.247	.147	.650	.496	.354	.581	.136	.082	.532	.153	.091	.582	.258	.168
DFCN [42]	Wechat	.653	.221	.137	.676	.487	.344	.701	.176	.114	.651	.193	.119	.670	.269	.179
Baseline	Wechat	.668	.076	.051	.655	.535	.431	.626	.170	.128	.670	.182	.152	.655	.241	.191
Ours	Wechat	**.823**	**.366**	**.252**	**.883**	**.727**	**.631**	**.764**	**.286**	**.214**	**.833**	**.405**	**.358**	**.826**	**.446**	**.364**

Table 1. Quantitative comparisons by using AUC, F1 and IoU as criteria. For each column within the same OSN transmission, the highest value is **bold**, and "-" indicates not applicable.

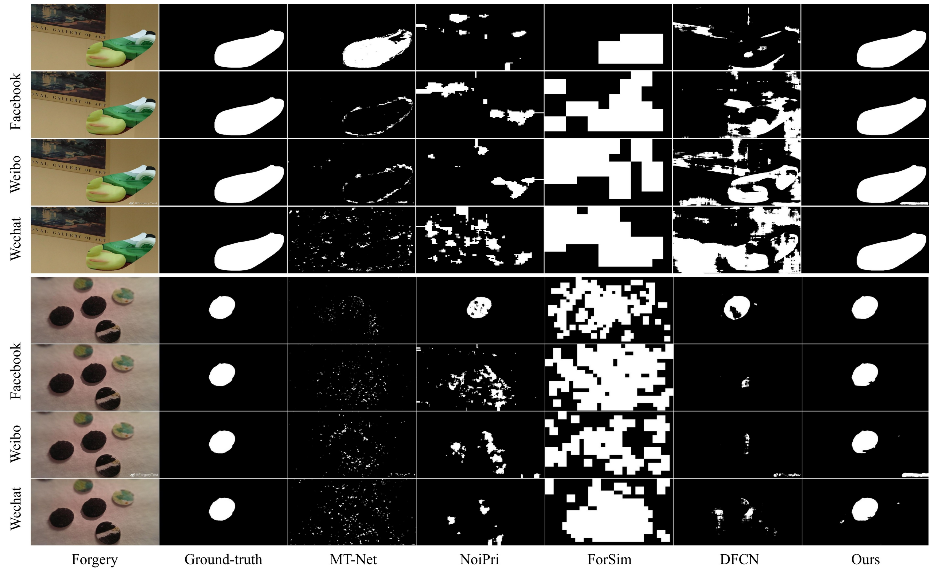

Figure 4. Qualitative comparisons for detecting the OSN-transmitted forgeries. For each row, the images from left to right are forgery (input), ground-truth, detection result (output) generated by **MT-Net** [37], **NoiPri** [12], **ForSim** [27], **DFCN** [42] and ours. The forgeries from top to bottom are the cases without OSN transmission, and with Facebook, Weibo and Wechat transmissions, respectively.

respectively, compared to the scenario without OSN transmission. In contrast, thanks to the appropriate noise modeling of $\boldsymbol{\tau}$ and $\boldsymbol{\xi}$, our proposed method exhibits rather desirable robustness against the OSN transmissions and still leads to accurate forgery detections. Taking Facebook for example, the IoU reduction is only 0.9%. It can also be noticed that the degradations of the forgery detection performance are slightly larger for Weibo and Wechat, with IoU reductions being 2.0% and 4.5%, respectively. This is mainly because, compared with Facebook, Weibo and Wechat adopt more stringent compressions for uploaded images, causing more evidence loss. In addition, for training our method, we only use the Facebook data, without any Weibo or Wechat data at all. From Tab. 1, we can see the scheme trained by using Facebook data can generalize well to Weibo and Wechat transmitted forgeries.

4.3. Qualitative comparisons

In addition to the quantitative comparisons, Fig. 4 gives two representative examples (see the supplementary file for more results). It can be seen that in the normal case (no OSN transmission), the existing detection methods perform

Detector f_{θ}	Test Datasets					
	Trans. w/o OSN			Trans. w/ Facebook		
	AUC	F1	IOU	AUC	F1	IOU
#1 SE-U-Net (baseline)	.745	.388	.294	.694	.331	.259
#2 SE-U-Net + τ	.755 (+.010)	.400 (+.012)	.325 (+.031)	.733 (+.039)	.377 (+.046)	.311 (+.052)
#3 SE-U-Net + ξ	.794 (+.049)	.471 (+.083)	.383 (+.089)	.753 (+.059)	.417 (+.086)	.340 (+.081)
#4 SE-U-Net + τ + ξ	.843 (+.098)	.496 (+.108)	.409 (+.115)	.847 (+.153)	.488 (+.157)	.400 (+.141)
#5 DPN	.719	.319	.224	.651	.208	.135
#6 DPN + τ + ξ	.778 (+.059)	.421 (+.102)	.350 (+.126)	.776 (+.125)	.449 (+.241)	.385 (+.250)

Table 2. Ablation studies regarding the modeling of τ and ξ. Values in brackets represent the differences with the baseline detector.

relatively well, *e.g.*, the **MT-Net** and **ForSim** in the first case, and the **NoiPri** and **DFCN** in the second case. However, these methods cannot achieve satisfactory detection performance in the cases of OSN transmitted versions. Take **NoiPri** in the second case for example. For Facebook, Weibo and Wechat transmitted images, the identified forged regions spread over several objects, making the forgery detection results much less useful. In contrast, our proposed method can learn more robust forgery features, and thereby generate more precise detection results over these challenging cases, primarily thanks to the robust training scheme with the compound noise modeling.

4.4. Ablation studies

We now conduct the ablation studies of our proposed training scheme by analyzing how each modelled noise (*i.e.*, the predictable noise τ and the unseen noise ξ) contributes to the final detection performance. To this end, we first prohibit the use of each noise in the scheme, and then evaluate the performance of different retrained detectors with appropriate settings. The obtained results are given in Tab. 2.

As can be seen, introducing the predictable noise τ in the training of the detector (#2 row) can slightly improve the detection performance (*e.g.*, 1.2% gains in F1), which is more obvious in the case of Facebook transmission (*e.g.*, 4.6% gains in F1). However, since it is incomplete to only adopt τ, as mentioned in Sec. 3.2, we further involve the designed unseen noise ξ. The results in #3 row imply that ξ can effectively enhance the robustness of the detector, bringing a more significant improvement (*e.g.*, 8.6% gains in F1). Finally, #4 row demonstrates that when the compound noise τ and ξ are applied simultaneously, the detector can be much more robust to the target environment, which is crucial for the forgery detection task over OSN transmission (*e.g.*, 15.7% gains in F1).

Further, instead of only using the SE-U-Net as the detector, we adopt another well-known architecture, DPN [10], to show the versatility of our proposed training scheme. As shown in rows #5 and #6, the robustness of the DPN can also be well strengthened by our robust training method.

4.5. Some further robustness evaluations

Although the proposed scheme is mainly designed to counter the lossy operations conducted by OSNs, we would also like to evaluate its robustness under some more commonly used degradation scenarios, such as noise addition, cropping, resizing, blurring, and standalone JPEG compression. Such evaluation is very critical in real-world cases because these types of post-processing operations are often adopted to erase or conceal the forged artifacts. To this end, we apply these post-processing operations to the original test set **Columbia** and report the quantitative comparisons in Fig. 5. For the convenience of demonstration, we utilize a unified parameter p for controlling the magnitudes of different operations. The origin of the horizontal axis ($p = 0$) corresponds to the case without any post-processing. As can be observed, the competitors [12,27,37,42] cannot perform consistently with the increase of the perturbation intensity, while our method can generalize well to defeat these post-processing operations.

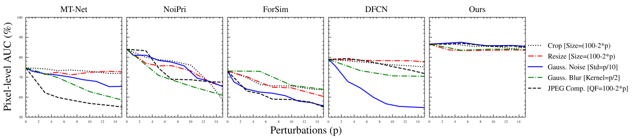

Figure 5. Robustness evaluations against cropping, resizing, blurring, noising and JPEG compression.

5. Conclusions

In this paper, we propose a novel training scheme for improving the robustness of the image forgery detection against various OSN-based transmissions. The proposed scheme is designed with the assistance of the modeling of a predictable noise τ as well as an intentionally introduced unseen noise ξ. Experimental results are provided to demonstrate the superiority of our scheme compared with several state-of-the-art methods. Further, we build an OSN-transmitted forgery dataset for the future research of the forensic community.

Acknowledgments: This work was supported by Macau Science and Technology Development Fund under SKL-IOTSC-2021-2023, 0072/2020/AMJ, 0015/2019/AKP, 060/2019/A1, and 077/2018/A2, by Research Committee at University of Macau under MYRG2018-00029-FST and MYRG2019-00023-FST, by Natural Science Foundation of China under 61971476, and by Alibaba Group through Alibaba Innovative Research Program.

References

[1] Nist nimble 2016 datasets. https://www.nist.gov/itl/iad/mig/nimble-challenge-2017-evaluation/. 2, 6, 7

[2] I. Amerini, T. Uricchio, L. Ballan, and R. Caldelli. Localization of jpeg double compression through multi-domain convolutional neural networks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn. Workshop*, 2017. 2

[3] Q. Bammey, R. Gioi, and J. Morel. An adaptive neural network for unsupervised mosaic consistency analysis in image forensics. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, pages 14194–14204, 2020. 2

[4] B. Bayar and M. C. Stamm. Constrained convolutional neural networks: a new approach towards general purpose image manipulation detection. *IEEE Trans. Inf. Forensics and Security*, 13(11):2691–2706, 2018. 1, 2

[5] L. Bondi, S. Lameri, D. Güera, P. Bestagini, E. J. Delp, and S. Tubaro. Tampering detection and localization through clustering of camera-based cnn features. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn. Workshops*, pages 1855–1864, 2017. 1, 2

[6] T. Carvalho, C. Riess, E. Angelopoulou, H. Pedrini, and A. Rezende Rocha. Exposing digital image forgeries by illumination color classification. *IEEE Trans. Inf. Forensics and Security*, 8(7):1182–1194, 2013. 2, 6, 7

[7] C. Chen, S. McCloskey, and J. Yu. Image splicing detection via camera response function analysis. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, pages 5087–5096, 2017. 2

[8] J. Chen, X. Kang, Y. Liu, and Z. J. Wang. Median filtering forensics based on convolutional neural networks. *IEEE Signal. Proc. Let.*, 22(11):1849–1853, 2015. 1, 2

[9] J. Chen, X. Liao, W. Wang, and Z. Qin. A features decoupling method for multiple manipulations identification in image operation chains. In *Proc. IEEE Int. Conf. Acoust. Speech Signal Process*, pages 2505–2509, 2021. 3

[10] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng. Dual path networks. In *Proc. Neural Info. Process. Syst.*, pages 4470–4478, 2017. 8

[11] D. Cozzolino, G. Poggi, and L. Verdoliva. Splicebuster: A new blind image splicing detector. In *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, pages 1–6, 2015. 1, 2

[12] D. Cozzolino and L. Verdoliva. Noiseprint: a cnn-based camera model fingerprint. *IEEE Trans. Inf. Forensics and Security*, 15(1):114–159, 2020. 1, 2, 6, 7, 8

[13] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Mach. Learn. Res*, 9(11), 2008. 5

[14] J. Dong, W. Wang, and T. Tan. Casia image tampering detection evaluation database. In *IEEE China Summit Inter. Conf. Signal Info. Proc.*, pages 422–426. IEEE, 2013. 2, 6, 7

[15] T. Gloe and R. Bohme. The dresden image database for benchmarking digital image forensics. *J. of Digit. Forensic Pract.*, 3(2-4):150–159, 2010. 6

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, pages 770–778, 2016. 4

[17] Y. Hsu and S. Chang. Detecting image splicing using geometry invariants and camera characteristics consistency. In *IEEE Inter. Conf. Multim. Expo*, pages 549–552. IEEE, 2006. 2, 6, 7

[18] M. Huh, A. Liu, A. Owens, and A. A. Efros. Fighting fake news: image splice detection via learned self-consistency. In *Proc. Eur. Conf. Comput. Vis.*, pages 101–117, 2018. 1, 2

[19] A. Islam, C. Long, A. Basharat, and A. Hoogs. Doa-gan: Dual-order attentive generative adversarial network for image copy-move forgery detection and localization. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, pages 4676–4685, 2020. 2

[20] X. Kang, M. C. Stamm, A. Peng, and K. R. Liu. Robust median filtering forensics using an autoregressive model. *IEEE Trans. Inf. Forensics and Security*, 8(9):1456–1468, 2013. 1, 2

[21] A. Li, Q. Ke, X. Ma, H. Weng, Z. Zong, F. Xue, and R. Zhang. Noise doesn't lie: towards universal detection of deep inpainting. In *Proc. Int. Jt. Conf. AI*, pages 1–7, 2021. 1, 2

[22] H. Li, W. Luo, and J. Huang. Localization of diffusion-based inpainting in digital images. *IEEE Trans. Inf. Forensics and Security*, 12(12):3050–3064, 2017. 1, 2

[23] Y. Li and J. Zhou. Fast and effective image copy-move forgery detection via hierarchical feature point matching. *IEEE Trans. Inf. Forensics and Security*, 14(5):1307–1322, 2019. 1, 2

[24] X. Liao, K. Li, X. Zhu, and K. J. Liu. Robust detection of image operator chain with two-stream convolutional neural network. *IEEE J. Sel. Top. Signal Process.*, 14(5):955–968, 2020. 3

[25] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft coco: common objects in context. In *Proc. Eur. Conf. Comput. Vis.*, pages 740–755, 2014. 6

[26] S. Lyu, X. Pan, and X. Zhang. Exposing region splicing forgeries with blind local noise estimation. *Int. J. Comput. Vis.*, 110(2):202–221, 2014. 1, 2

[27] O. Mayer and M. C. Stamm. Forensic similarity for digital images. *IEEE Trans. Inf. Forensics and Security*, 15(1):1331–1346, 2020. 1, 2, 6, 7, 8

[28] O. Ronneberger, P. Fischer, and T. Brox. U-net: convolutional networks for biomedical image segmentation. In *Proc. Int. Conf. Med. Image Comput. Computer-Assisted Int.*, pages 234–241. Springer, 2015. 4, 6

[29] A. G. Roy, N. Navab, and C. Wachinger. Recalibrating fully convolutional networks with spatial and channel "squeeze and excitation" blocks. *IEEE Trans. Medical Imaging*, 38(2):540–549, 2018. 6

[30] S. Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016. 5

[31] R. Shin and D. Song. Jpeg-resistant adversarial images. In *Proc. Neural Info. Process. Syst. Workshop*, pages 1–6, 2017. 4

[32] W. Sun, J. Zhou, L. Dong, J. Tian, and J. Liu. Optimal prefiltering for improving facebook shared images. *IEEE Trans. Image Process.*, 30(1):6292–6306, 2021. 2

[33] W. Sun, J. Zhou, Y. Li, M. Cheung, and J. She. Robust high-capacity watermarking over online social network shared im-

ages. *IEEE Trans. Circuits Syst. Video Technol.*, 31(3):1208–1221, 2020. 2, 3, 6

[34] W. Sun, J. Zhou, R. Lyu, and S. Zhu. Processing-aware privacy-preserving photo sharing over online social networks. In *ACM Int. Conf. Multimedia*, pages 581–585. ACM, 2016. 2

[35] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and B. Fergus. Intriguing properties of neural networks. In *Proc. Int. Conf. Learn. Representat.*, pages 1–10, 2014. 2, 5

[36] H. Wu and J. Zhou. Iid-net: image inpainting detection network via neural architecture search and attention. *IEEE Trans. Circuits Syst. Video Technol.*, pages 1–14, 2021. 1, 2

[37] Y. Wu, W. AbdAlmageed, and P. Natarajan. Mantra-net: manipulation tracing network for detection and localization of image forgeries with anomalous features. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, pages 9543–9552, 2019. 1, 2, 6, 7, 8

[38] J. You, Y. Li, J. Zhou, Z. Hua, W. Sun, and X. Li. A transformer based approach for image manipulation chain detection. In *ACM Int. Conf. Multimedia*, pages 3510–3517. ACM, 2021. 3

[39] J. Zhong and C. Pun. An end-to-end dense-inceptionnet for image copy-move forgery detection. *IEEE Trans. Inf. Forensics and Security*, 15(1):2134–2146, 2020. 1, 2

[40] P. Zhou, X. Han, V. Morariu, and L. Davis. Two-stream neural networks for tampered face detection. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn. Workshop*, 2017. 2

[41] P. Zhou, X. Han, V. I. Morariu, , and L. S. Davis. Learning rich features for image manipulation detection. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, pages 1907–1915, 2018. 2

[42] P. Zhuang, H. Li, S. Tan, B. Li, and J. Huang. Image tampering localization using a dense fully convolutional network. *IEEE Trans. Inf. Forensics and Security*, 16(1):2986–2999, 2021. 1, 2, 6, 7, 8