EURASIP Journal on Image and Video Processing

RESEARCH Open Access

Defeating data hiding in social networks using generative adversarial network

Huaqi Wang[1], Zhenxing Qian[2*], Guorui Feng[1] and Xinpeng Zhang[2]

* Correspondence: zxqian@fudan.edu.cn
[2]Shanghai Institute of Intelligent Electronics and Systems, School of Computer Science, Fudan University, Shanghai, China
Full list of author information is available at the end of the article

Abstract

As a large number of images are transmitted through social networks every moment, terrorists may hide data into images to convey secret data. Various types of images are mixed up in the social networks, and it is difficult for the servers of social networks to detect whether the images are clean. To prevent the illegal communication, this paper proposes a method of defeating data hiding by removing the secret data without impacting the original media content. The method separates the clean images from illegal images using the generative adversarial network (GAN), in which a deep residual network is used as a generator. Therefore, hidden data can be removed and the quality of the processed images can be well maintained. Experimental results show that the proposed method can prevent secret transmission effectively and preserve the processed images with high quality.

Keywords: Information hiding, Social networks, Steganography, Steganalysis

1 Introduction

With the fast development of information technology, the online social networks (OSN) can provide us a convenient transmission of various messages. However, terrorists can also use OSN to transmit secret messages by hiding data inside the posted images. Generally, it is difficult for a server to detect whether an image contains secret messages inside the content. One possible solution is to interfere with the image content in OSN and destroy the hidden data that might be embedded.

There are two categories of data hiding technologies, i.e., steganography and watermarking [1]. The former hides many data into a cover while aiming at avoiding detection. In most cases, steganography is fragile to common attacks, and hidden data can be removed easily. The latter focuses on embedding data robustly, making the hidden data difficult to be destroyed. However, fewer data can be hidden into a cover by watermarking, which is widely used for copyright protection in social networks [2, 3]. Steganalysis is a technique to detect whether an image contains hidden data [4, 5]. However, steganalysis is not precise enough, esp. in cases of small embedding rates [6]. Besides, as there are many processed images in OSN, it would inevitably necessarily result in large false alarm rates. Therefore, it is more reliable to defeat the covert transmission by interfering with the image content.

Typical image processing operations on images, e.g., recompression, down-sampling, and beautification [7], can defeat most steganography methods without robustness. However, it is difficult to remove the messages hidden by robust steganography or watermarking tools. In previous studies, researchers have proposed some methods of destroying digital watermarking. In [8], an attacking method is proposed to remove redundancy through the self-similarities of image pixels. In [9, 10], the wavelet transform-based watermarking and singular value decomposition-based watermarking can be defeated, respectively. These methods are mainly useful for specific watermarking algorithms [11].

The development of deep learning brings forward more tools for image processing, e.g., image classification, reconstruction, and recognition [12–15]. As most data-hiding methods can be viewed as adding noises, it would be useful to remove the hidden data by image denoising. Although many methods in [16–20] can offer better denoising performances than traditional methods, they are not good at removing the hidden data, esp. the date hidden by robust information-hiding tools. In this paper, we propose a new framework of defeating covert transmission in OSN. Inspired by the generative adversarial network (GAN) [21], we design a generator and a discriminator to destroy the secret data that might be hidden in the OSN images. After processing the images using the proposed method, a receiver cannot extract the hidden data from the processed images, even the robust data hiding methods are used by the sender. Meanwhile, the image quality can be well preserved. The rest of the paper is organized as follows. Section 2 introduces the related works. In Section 3, we present a detailed implementation of the proposed framework. Experimental results are presented in Section 4, and Section 5 concludes the paper.

2 Related works

Social networks such as Weibo, Twitter, and Instagram have various image processing functions. General steganography algorithms are not robust, in other words, social networks can easily break the secret information of stego images. Therefore, we use watermarking algorithms to verify the performance of our method, considering that terrorist may apply the robust and imperceptible watermarking for covert communication. The algorithms for testing should be typical and will not fail to normal lossy channel. After comprehensive consideration, we chose three classic algorithms in the field of digital image watermarking, which are based on quantized index modulation (QIM) [22], spread spectrum (SS) [23], and uniform log-polar mapping (ULPM) [24], respectively. Some brief introductions are given in this section.

The QIM algorithm quantifies the original cover into several different index intervals by different quantifiers, which is also the embedding process. There are generally two quantifiers due to the embedded information that is binary, and the quantization area is not coincident because of the disjunction. The watermarking will be extracted according to the quantitative index interval of modulated data. Receiver can detect hidden data by the shortest distance method when the channel interference is not serious.

As an important work in frequency domain watermarking, the contribution of SS algorithm lies in the introduction of spread spectrum communication technology. The spread spectrum code with pseudorandom and cross-correlation properties plays a key

role in system. And the energy distribution of embedded watermarking signal is extended to a wider spectrum, which improves the security and robustness capability.

The researchers of ULPM algorithm propose a watermarking robust to rotated, scaled, translated, cropped distortion and general print-scan simultaneously. This study eliminates the interpolation distortion and expands the embedding space. A discrete log-polar point can be obtained by performing the near ULPM to the frequency index in the Cartesian system, and the data of which is then embedded to the corresponding DFT coefficient in Cartesian system, ensuring the integrated robust performance and efficiency.

Although the above three watermarking algorithms have difference in robustness, they will not easily fail in the face of lightweight image processing in social networks. Our method can prevent illegal communication by using of robust watermarking, so as to break the hidden data that cannot be influenced by traditional attacks. And the quality of processed images is slightly affected. Meanwhile, the method can be regarded as a new evaluation for the robustness of information hiding.

3 Proposed method

3.1 Overall framework

The flowchart of the proposed method is illustrated in Fig. 1. We provide a holistic approach to prevent security risks in social networks, which no longer relies on steganalysis due to possible failure in detection. We first generate watermarked image sets by randomly adding watermark into normal images, and we use binary random sequences as secret data, i.e., possibilities for 0 and 1 are equal. We send all pairs of image sets to GAN and gain the processed models by learning the mapping of watermarked images to clean images. Subsequently, all models would be integrated into the

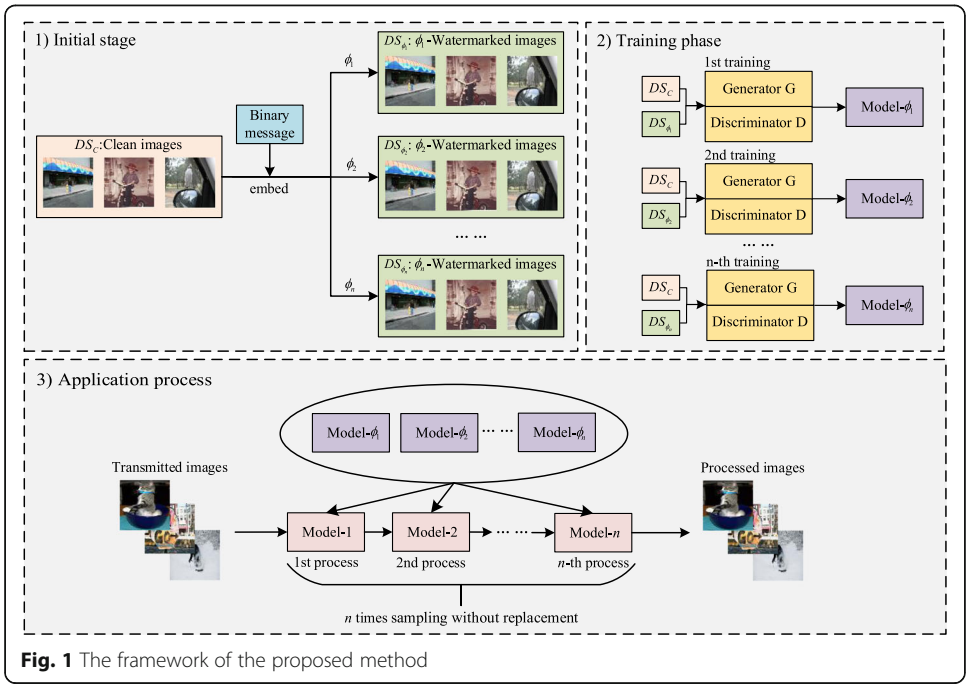

Fig. 1 The framework of the proposed method

social networks to block illegal communication hidden in transmitted images. The detailed steps are described as follows:

1) In the initial stage, clean images of the database DS_C are first added with a watermark generated by random binary message. We denote the watermarking algorithms as ϕ_1, ϕ_2, …, ϕ_n respectively, and the corresponding watermarked image datasets as $DS_{\phi_1}, DS_{\phi_2}, \ldots, DS_{\phi_n}$. The watermarking algorithms should cover both classical and state-of-the-art algorithms.
2) In the training phase, DS_C is sent to the generator G and discriminator D n times with n watermarked datasets severally. We follow the optimization equation proposed by [21]

$$\min_G \max_D V(D,G) = E_{p_{\text{data}}}(x)[\log D(x)] + E_{p_z}(z)[\log(1 - D(G(z)))] \tag{1}$$

where $p_{\text{data}}(x)$ and $p_z(z)$ denote the distribution of real data and generated false data, respectively. E calculates their mathematical expectation. The value function V represents the performance of D. For each training objective, G fits from the prior distribution on DS_C, ensuring that the expected error of D for the generated data is as large as possible. Then D should distinguish the real samples from the generated samples more accurately through the log-likelihood. The Model-ϕ_1, Model-ϕ_2,…, Model-ϕ_n record training parameters for each session. The details of the network design will be introduced in Sections 3.2 and 3.3.

3) We deploy all the aforementioned training models on social networks in the application process. One of the rules is not to judge whether a transmitted image contains a watermark and the type of watermark to guarantee the practicality of our method. The effect of each model is only valid for images with its corresponding or similar watermarking algorithms due to the characteristic of data distribution. Therefore, we scramble all models under random n times sampling without replacement for n times process, and the rearrangement is Model-1, Model-2,…, Model-n. The operation would be done whenever an image is transmitted. As an example, the Model-ϕ_1 gained by DS_C and DS_{ϕ_1} should have a small influence on the ϕ_2-watermarked image. However, the watermarked image that applies ϕ_2 can be processed by Model-ϕ_2 during n times process to remove secret data in any case.

It should be noted that our training scheme is not to mix the watermarked images of all labels. Because different types of watermarked images have great differences in data distribution, it may lead to the instability of network learning and the failure of the models. The framework avoids the problem to some extent. Meanwhile, it is apparent that the data distribution of clean images is different from that of the watermarked images, which also guarantees that clean images are not largely affected. We obtain the result under n times random sampling to ensure the randomness of the processed image at the pixel level. Besides, in many cases, data senders are able to find patterns of

image processing in social networks by repeatedly uploading and downloading. The framework prevents such phenomenon effectively.

3.2 The architecture of generator G

We use the method in [18] as the generator to gain the mapping of watermarked images to clean images. The applied convolutional neural network (CNN) can efficiently and flexibly mine deep features of images by combining residual learning and batch normalization (BN). Because of the truth that the deeper networks generated by merely adding layers would not always bring positive benefits, the combination method avoids convergence difficulties and the saturation or even slowdown in network performance.

We synchronize the training errors of deep and shallow networks by introducing shortcut connections on the stacked layer. Specifically, we denote the original mapping to be learned as $\mathcal{H}(x)$ for network input x and output y, while the residual mapping is $\mathcal{F}(x) = \mathcal{H}(x) - x$. When the residual is zero, the network would not be negatively optimized because the identity mapping happens on the stack layer. In theory, the most intuitive benefit is to cut the amount of learning required to make training more accessible. Next, we take the residual image as output directly through only one residual unit, which is different from the classic residual network with multiple shortcut connections. At the same time, the BN layer is employed to improve the generalization ability and reduce the training pressure caused by adapting to the distribution changes of each iteration.

Figure 2 provides the architecture of generator and discriminator network. The network depth is set to 21, which is determined by balancing model effect and training time. We apply 64 filters of size 3 × 3 on the input watermarked image I^W. The output 64 feature maps are fed into the 19 repeated convolutional layers composed of 64 kernels with size 3 × 3, and batch normalization is added after each convolution. The

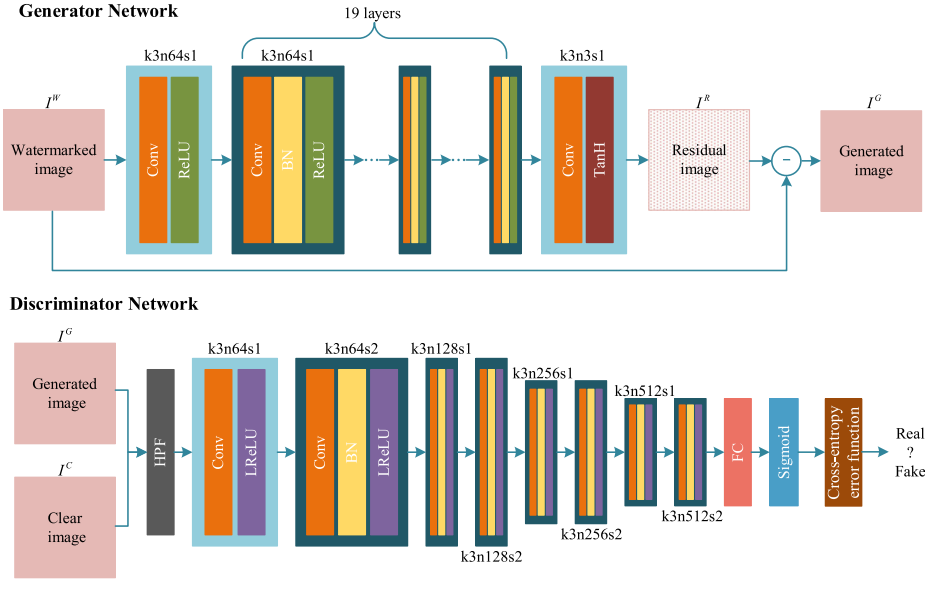

Fig. 2 Architecture of generator and discriminator network with kernel size (k), number of feature maps (n), and stride (s) represent the parameters of each convolutional layer

residual image I^R is reconstructed by the corresponding number of image channels, aiming to approximate the real residual of I^W and clean image I^C. Except for the TanHyperbolic (TanH) function used on the output layer, all other layers take rectified linear units (ReLU) as the activation function for the stability of training. At the end of the network, generated image I^G is obtained by subtracting I^R from I^W. We denote training parameters of the generator G as $\theta_G = \{\omega_{1\sim L}; b_{1\sim L}\}$, where $\omega_{1\sim L}$ and $b_{1\sim L}$ represent the weights and biased of the *L-th* layer, respectively. We express the relationship between the above image labels by Eq. (2).

$$\begin{cases} I^R = G_{\theta_G}(I^W) \\ I^G = I^W - I^R \end{cases} \tag{2}$$

We use a real-valued tensor of size $N\times H\times W\times C$, where the images are sized $N\times H\times W$ with C channels, and the training batch size is N.

Our learning goal is guided by the loss function, which consists of content loss and adversarial loss. The content loss adopts the mean-squared error (MSE) of the output residual image and the real residual as the optimization objective, which is the most frequently used in the perceptual loss. Since it can be intuitively regarded as the pixel-wise difference, the detailed result is calculated by Eq. (3)

$$l_{mse} = \frac{1}{NHW}\sum_{k=1}^{N}\sum_{m=1}^{H}\sum_{n=1}^{W}\left(I^C_{k,m,n} - \left(I^W_{k,m,n} - I^R_{k,m,n}\right)\right)^2 \tag{3}$$

However, the accuracy of gradient descent direction is not high enough by simply using error back-propagation through MSE, especially where there is little visual disparity between watermarked image and target clean image. We expect that the probability of a fake image being judged as clean by discriminator is vast, and keep pace with the minimization trend of MSE. Therefore, the adversarial loss is further added to update gradient more precisely and make sure the generated image is as similar as possible to the groundtruth. The adversarial loss can be calculated as follows:

$$l_{adv} = \frac{1}{N}\sum_{k=1}^{N} - \log D_{\theta_D}\left(I^G_k\right) \tag{4}$$

Finally, we define the total generator loss as

$$l_G = l_{\text{mse}} + \beta l_{\text{adv}} \tag{5}$$

where $\beta = 10^{-3}$. Empirically, for the balance of generator and discriminator, the proportion of adversarial loss is generally slightly smaller.

3.3 The architecture of discriminator D

We set up a pre-processing layer based on prior knowledge before the image is formally inputted into the discriminator. Image quality would affect the results of an algorithm under normal circumstances. The processing of database is not restricted to the normalization of image pixels. It is crucial to eliminate irrelevant information and take advantage of useful information on the basis of simplifying data to the greatest extent. Because the difference between watermarked image and clean image is totally small in our task, it can be regarded as a weak noise signal in high frequency. High-pass filtering operation can amplify the signal by weakening the other image contents, which would

drive the subsequent network to perform better at classification. We denote the high-pass filter as F, and the filtered image R under batch N can be obtained by Eq. (6)

$$R_k^{\text{label}} = I_k^{\text{label}} \otimes F \tag{6}$$

where $k = 1, 2, \ldots, N$. The symbol $\otimes$ represents convolution operation, and the label on behalf of generated image G and clean image C. We use the following filter kernel, which is commonly employed in steganalysis.

$$K_{HPF} = \frac{1}{12}\begin{pmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{pmatrix} \tag{7}$$

Inspired by the principles summarized in DCGAN [25], the core part of the discriminator network consists of 8 convolutional layers, and the number of kernels increases gradually from 64 to 512 by a factor of 2. We utilize the stacked convolution kernel of size 3 × 3 instead of the size 5 × 5 used in the original method without changing the perceptive field. This setting allows the mapping to contain more nonlinear functions and to represent more features with fewer parameters. The probability of sample classification is calculated by the cross-entropy error function, after 512 feature maps pass through the full connection layer and sigmoid activation function. More importantly, we add the BN layer and the LeakyReLU activation function in all convolutional layers except the input layer for the sake of discrimination stability.

Similarly, we utilize parameter θ_D to construct discriminator as D_{θ_D}. The optimization goal is defined as follows:

$$\max_D E_{I^C \sim p_{\text{train}}(I^C)} \log D_{\theta_D}(I^C) + E_{I^G \sim p_G(I^G)}\left(1 - \log D_{\theta_D}(I^G)\right) \tag{8}$$

The discriminator is able to determine the probability of real as higher as possible when the input image is clean. For the generated fake image, the detecting result is low. The network achieves Nash equilibrium during the interaction between discriminator and generator, and the final generated image is sufficient to deceive discriminator.

4 Results and discussion

4.1 Experimental setting

We test three classic watermarking algorithms based on QIM, SS, and ULPM, respectively. The image dataset employed in our experiments is COCO [26], which contains 200,000 plain color images. In practice, we select 10,000 images from training set and 1000 images from testing set randomly for experiments. A larger training naturally will increase the computational complexity and might cause positive feedback to the results. All images are resized to 192 × 192 for simplicity.

In the initial stage before training, we first set the label of the original training image as clean. Next, the above-mentioned three watermarking algorithms are utilized to generate watermarked image denoted as ϕ_{QIM}, ϕ_{SS}, and ϕ_{ULPM}. The length of message sequence is randomly selected from 40-bit to 120-bit. According to the payload capacity of each algorithm, we consider the length range of message comprehensively, which enlarges the effect of the model on watermarked images with various data extent. Though these watermarking algorithms are mainly designed for gray images, they can be easily

applied on color images by embedding data in the Y channel. We separately send the clean images and three watermarked datasets to GAN to gain three processed models named Model-ϕ_{QIM}, Model-ϕ_{SS}, and Model-ϕ_{ULPM}.

The image pre-processing of network includes normalizing the pixels to [-1, 1] and high-pass filtering. Our models are trained for 7500 iterations based on the Adam optimizer, and hyperparameter momentum is set to 0.9. The learning rate is decayed exponentially from 1e−4 to 1e−6. To avoid the oscillation of loss, all weights are initialized by a normal distribution with a mean of 0 and a standard deviation of 0.02. The slope is 0.2 in all layers activated by Leaky ReLU. We conduct the experiments on a PC with Intel (R) Core (TM) i7-6850K CPU 3.60 GHz and a GTX1080Ti GPU. It averagely takes about 1.5 days to train each model on GPU.

4.2 Evaluations on process effectiveness

For objective image assessment, we use three metrics to assess the degree of damage and the impact on the quality of watermarked images. The value of each objective metric is the mean result on testing sets. The first is the data extraction error rate of processed images. We denote the number of wrong message bits as n_{error}, and n_m is the length of embedded messages, the error rate result is calculated by Eq. (9)

$$R_{\text{error\_rate}} = \frac{n_{\text{error}}}{n_m} \times 100\% \tag{9}$$

which approaches 50% means that secret data is completely destroyed. Peak signal-to-noise-ratio (PSNR) and structural similarity index (SSIM) as two universal criteria are also applied. The former measures fidelity of watermarked images and processed images, while the latter evaluates visual loss. A higher PSNR or SSIM generally indicated better visual quality.

We test the effectiveness of each processed model in the first step to ensure that the saved models can process corresponding watermarked images. The lengths of secret message are 40, 60, 80, 100, and 120 bits, respectively. As in the training phase, the message is also embedded in the Y channel. Figure 3 shows the relationship between

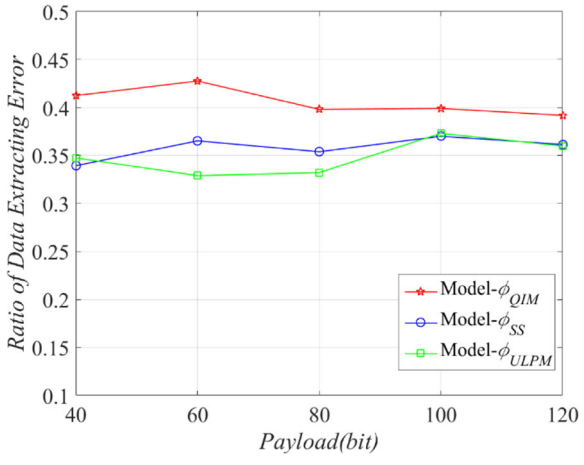

Fig. 3 Relationship between data extracting error and payload 40, 60, 80, 100, and 120 bits on Model-φ_{QIM}, Model-φ_{SS}, and Model-φ_{ULPM}

data extracting errors and payloads. For the testing images of ϕ_{QIM} watermark scheme, the average error rate can reach around 40% or higher, which indicates the secret data has been basically destroyed. While the watermarked images of ϕ_{SS} and ϕ_{ULPM} perform slightly better than ϕ_{QIM} in fault tolerance due to non-blind and error-correction code. However, the ratios of data error for each payload tested are more than 30%, indicating that the extracted data has lost the original meaning.

Figure 4 shows the effect of model on the quality of watermarked images. With payload increasing, the influence of Model-ϕ_{QIM} and Model-ϕ_{SS} is getting larger, and Model-ϕ_{ULPM} is stabilizing. However, high SSIM proves strong imperceptibility of the proposed framework. As we reconstruct the pixel content of watermarked images to approximate their original images, the degree of impact on image quality depends on the watermark algorithm principle.

As mentioned above, it is meaningless to apply a single model to the images watermarked by the corresponding algorithm in practice because we cannot classify the type of transmitted images. Hence, we further serial all models in random order so that images are processed three times. Obviously, there are six kinds of outcomes. We denote all processes as $P_{\mathrm{QIM-SS-ULPM}}$, $P_{\mathrm{QIM-ULPM-SS}}$, $P_{\mathrm{SS-QIM-ULPM}}$, $P_{\mathrm{SS-ULPM-QIM}}$, $P_{\mathrm{ULPM-QIM-SS}}$, and $P_{\mathrm{ULPM-SS-QIM}}$. Next, we embed 80-bit and 100-bit messages in the images of testing set by ϕ_{QIM}, ϕ_{SS}, and ϕ_{ULPM} to generate the watermarked images sets named $T^{80}_{\phi_{\mathrm{QIM}}}$, $T^{80}_{\phi_{\mathrm{SS}}}$, $T^{80}_{\phi_{\mathrm{ULPM}}}$, $T^{100}_{\phi_{\mathrm{QIM}}}$, $T^{100}_{\phi_{\mathrm{SS}}}$, and $T^{100}_{\phi_{\mathrm{ULPM}}}$.

Further, toward better proof for the performance of our method, we also test the same metrics on watermarked images processed by several traditional distortions, including JPEG compression, gamma correction, Gaussian noise, salt and pepper noise, wiener filtering, Gaussian filtering, and median filtering. We set the quality factor of JPEG compression QF= 90, 70, 50, 30, and 10. For other kinds of attacks, the filtering window size is 5 × 5, mean and variance of noise are 0 and 0.05, and the gamma factor is 0.3.

To demonstrate the superiority of our method over the traditional attacks in visual, we select the label "test_image22.png" in the testing sets as an object, and the version of ϕ_{SS} with a 100-bit message is "ϕ^{100}_{SS}_image22.png". Subsequently, we give all processed images of the "ϕ^{100}_{SS}_image22.png" applied by our method and the above traditional attacks, which is shown in Fig. 5. As can be seen from the results, our

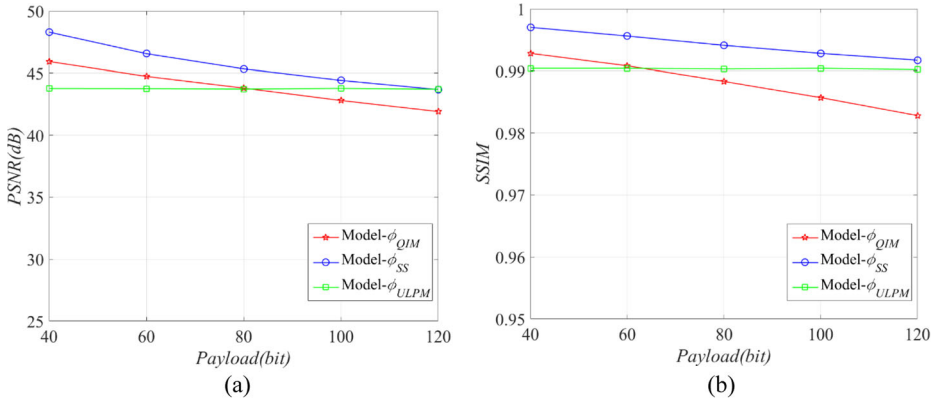

Fig. 4 Image quality of watermarked images with payload 40, 60, 80, 100, and 120 bits processed by Model-φ_{QIM}, Model-φ_{SS}, and Model-φ_{ULPM} for **a** PSNR and **b** SSIM

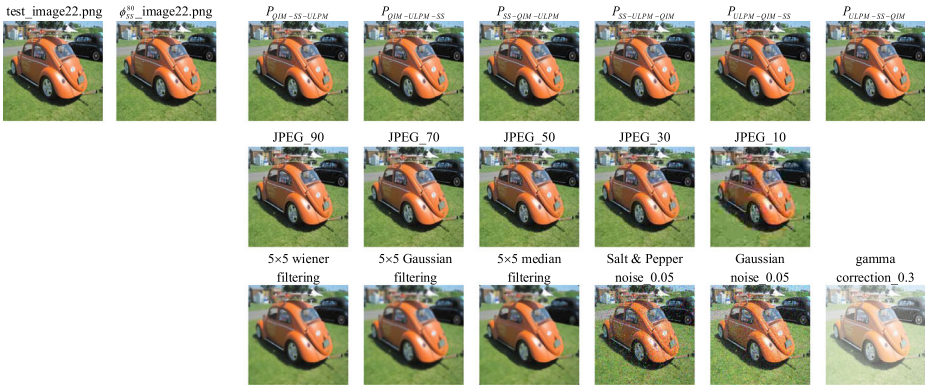

Fig. 5 Processed images of "φ_{SS}^{80}_image22.png" applied by our method, JPEG compression, wiener filtering, Gaussian filtering, median filtering, salt and pepper noise, Gaussian noise, and gamma correction

processed images are almost identical to that before processing. However, with the decrease of the quality factor, the perception of visual distortion increases gradually. Meanwhile, the results of image filtering, noising, and gamma correction are obviously not promising.

We compare the recovery of the 80-bit message and image quality variation in the six processes with JPEG compression. The results of all outcomes are shown in Table 1. It is observed that different order of three models offers individual results. For a watermarked image, the best situation is that the model trained by the corresponding watermarked images is placed first. Other models produce incorrect effects toward a clean image in pixel content to bring a chain reaction. JPEG compression as the most common image processing operation works explicitly until the QF= 10. However, the image quality will drastically deteriorate, which is not allowed in real social network application. The worst results from randomization in our method can also ensure channel security without much change in image quality.

Other traditional attacks mentioned above and the six processes are applied on watermarked images with a 100-bit message, and testing results are listed in Table 2.

Table 1 Comparisons of JPEG compression and the proposed method in error rate and image quality for watermarked images with payload 80 bits

	$T_{\varphi_{QIM}}^{80}$			$T_{\varphi_{SS}}^{80}$			$T_{\varphi_{ULPM}}^{80}$		
	Error rate	PSNR	SSIM	Error rate	PSNR	SSIM	Error rate	PSNR	SSIM
$P_{QIM-SS-ULPM}$	**43.55%**	**41.61**	**0.9860**	**29.54%**	**44.03**	**0.9928**	**33.54%**	**43.40**	**0.9901**
$P_{QIM-ULPM-SS}$	**43.46%**	**41.65**	**0.9860**	**22.67%**	**44.40**	**0.9932**	**37.44%**	**43.53**	**0.9902**
$P_{SS-QIM-ULPM}$	**33.64%**	**41.49**	**0.9851**	**37.19%**	**43.84**	**0.9924**	**37.56%**	**43.40**	**0.9901**
$P_{SS-ULPM-QIM}$	**49.82%**	**41.46**	**0.9851**	**26.46%**	**44.22**	**0.9930**	**37.41%**	**43.55**	**0.9902**
$P_{ULPM-QIM-SS}$	**34.68%**	**41.52**	**0.9844**	**37.26%**	**43.85**	**0.9925**	**37.58%**	**43.41**	**0.9901**
$P_{ULPM-SS-QIM}$	**37.61%**	**41.45**	**0.9842**	**31.73%**	**43.61**	**0.9918**	**37.38%**	**43.55**	**0.9902**
JPEG_90	0.11%	34.94	0.9756	0.00%	34.94	0.9754	0.00%	34.90	0.9748
JPEG_70	2.34%	31.62	0.9523	0.00%	31.62	0.9519	5.95%	31.52	0.9491
JPEG_50	2.27%	30.17	0.9359	0.00%	30.18	0.9356	14.36%	30.06	0.9308
JPEG_30	17.94%	28.77	0.9136	0.30%	28.74	0.9199	21.68%	28.67	0.9068
JPEG_10	50.96%	25.40	0.8199	43.36%	25.43	0.8209	49.18%	25.40	0.8168

Table 2 Comparisons of Wiener filtering, Gaussian filtering, median filtering, salt and pepper noise, Gaussian noise and gamma correction, and the proposed method in error rate and image quality for watermarked images with payload 100 bits

	$T^{100}_{\varphi_{QIM}}$			$T^{100}_{\varphi_{SS}}$			$T^{100}_{\varphi_{ULPM}}$		
	Error rate	PSNR	SSIM	Error rate	PSNR	SSIM	Error rate	PSNR	SSIM
$P_{QIM-SS-ULPM}$	**44.24%**	**40.86**	**0.9834**	**29.34%**	**43.30**	**0.9914**	**37.38%**	**43.46**	**0.9902**
$P_{QIM-ULPM-SS}$	**44.23%**	**40.89**	**0.9834**	**22.39%**	**43.67**	**0.9919**	**37.37%**	**43.59**	**0.9903**
$P_{SS-QIM-ULPM}$	**35.09%**	**40.72**	**0.9822**	**38.00%**	**43.16**	**0.9911**	**37.48%**	**43.46**	**0.9902**
$P_{SS-ULPM-QIM}$	**49.95%**	**40.69**	**0.9823**	**26.58%**	**43.50**	**0.9916**	**37.38%**	**43.61**	**0.9903**
$P_{ULPM-QIM-SS}$	**36.19%**	**40.74**	**0.9814**	**37.96%**	**43.17**	**0.9911**	**37.41%**	**43.48**	**0.9902**
$P_{ULPM-SS-QIM}$	**38.84%**	**40.66**	**0.9811**	**34.17%**	**42.84**	**0.9901**	**37.39%**	**43.61**	**0.9903**
5 × 5 Wiener filtering	9.21%	29.22	0.8817	15.92%	29.37	0.8930	19.93%	29.22	0.8840
5 × 5 Gaussian filtering	14.13%	25.41	0.8479	19.26%	25.48	0.8543	20.38%	25.42	0.8443
5 × 5 median filtering	3.93%	22.63	0.7263	3.34%	33.65	0.7291	18.20%	22.63	0.7247
Salt and pepper noise_0.05	31.19%	17.92	0.5156	19.45%	17.92	0.5135	28.45%	17.97	0.5183
Gaussian noise_0.05	22.61%	19.45	0.5664	12.74%	19.45	0.5640	24.21%	19.46	0.5663
gamma correction_0.3	5.31%	10.39	0.6747	7.28%	10.39	0.6769	0.00%	10.39	0.6752

Although different watermarking algorithms have different performance in resisting various kinds of traditional attacks, the data extraction error rate is still inferior to our method while the watermarked images have been seriously distorted, according to Fig. 5. We can speculate that the traditional attacks will cause intolerable distortion to watermarked images when achieving sufficient data error rate, which further proves the effectiveness of our proposed method.

4.3 Anti-analyzability of process and impact on clean images

Our framework provides randomness from different order of sampling of the models, so that robustness against collusion attack is ensured, namely, finding the inherent rule and designing resistance strategy. To illustrate randomness, we should prove that each process order has different effect results on an image. The "test_image616.png" and "ϕ^{100}_{ULPM}_image616.png" are selected from respective image sets. Six processes are applied

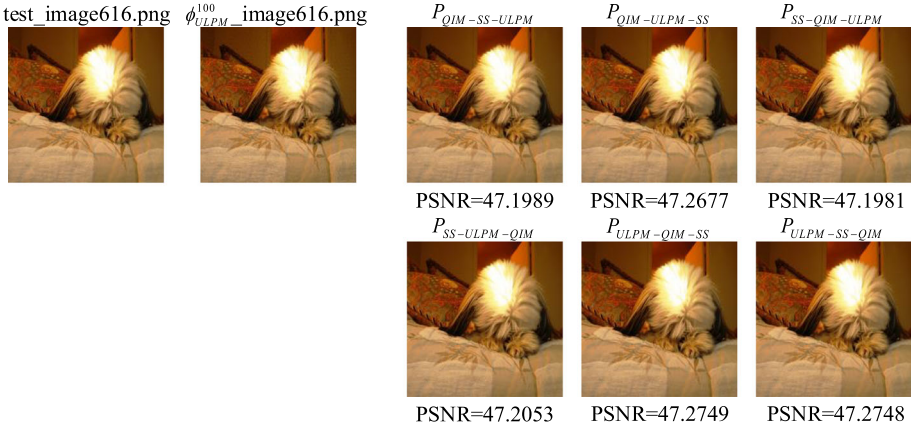

Fig. 6 The "φ^{100}_{ULPM}_image616.png" and different processed consequences

Table 3 Impact on clean images

	$P_{QIM-SS-ULPM}$	$P_{QIM-ULPM-SS}$	$P_{SS-QIM-ULPM}$	$P_{SS-ULPM-QIM}$	$P_{ULPM-QIM-SS}$	$P_{ULPM-SS-QIM}$
PSNR	49.65	49.76	49.65	49.78	49.67	49.79

to the watermarked image. The PSNR between the processed image and the watermarked image is used to distinguish outcomes. The results are shown in Fig. 6. Apart from the fact that human eyes can barely distinguish differences, we assure that the distribution of internal pixels is different through image quality.

On the other hand, the majority of images transmitted over social networks are free from secretly embedded data. It is also necessary to verify that the model has little effect on these pure images. We process pure images without any watermarking in testing sets using the six processes in Section C and list the average value of PSNR and SSIM of these images in Table 3. The results in Table 3 prove that the impact of defeating potential data hiding proposed in the paper is mere and controllable. Also, better performance of removing secret data will result in lower influence on the non-watermarked images.

5 Conclusion

In the paper, we consider that social networks are weak in the face of illegal communication hidden by robust algorithms, and steganalysis performs not well in the small payload. We propose a GAN-based method to defeat data hiding, which learns the mapping from the watermarked images to the corresponding clean images. The experiments prove that the process models trained are effective in destroying hidden data basically while ensuring the quality of the processed image. To resist collusion attack, we increase the vigilance for communication channel analysts by sampling without replacement repeatedly from the process models. For future study, we consider to improve the breaking rate and integrate more robust data hiding schemes by designing more efficient schemes to integrate all watermarking algorithms.

Abbreviations

GAN: Generative adversarial network; OSN: Online social networks; CNN: Convolutional neural network; BN: Batch normalization; TanH: TanHyperbolic; ReLU: Rectified linear units; QIM: Quantized index modulation; SS: Spread spectrum; ULPM: Uniform log-polar mapping; PSNR: Peak signal-to-noise-ratio; SSIM: Structural similarity index

Acknowledgements

Thanks to the anonymous reviewers for their constructive suggestions to help improve this paper.

Authors' contributions

The first author (HW) participated in the designing of the method, carried out the experiments, and composed the manuscript. The second author (ZQ) conceived of the study, participated in the design, and helped to draft the manuscript. The third author (GF) and the fourth author (XZ) helped to design and improve the method. All authors read and approved the final manuscript.

Authors' information

Huaqi Wang received B.S. degree from Shanghai University, China, in 2018, where Wang is currently pursuing M.S. degree. Her research interests include information hiding and multimedia security.

Zhenxing Qian received B.S. and Ph.D. degrees from the University of Science and Technology of China (USTC), in 2003 and 2007, respectively. He is currently a professor with the School of Computer Science, Fudan University. He has published over 100 peer-reviewed papers on international journals and conferences. His research interests include information hiding, image processing, and multimedia security.

Guorui Feng received B.S. and M.S. degree in computational mathematics from Jilin University, China, in 1998 and 2001, respectively. He received Ph.D. degree in electronic engineering from Shanghai Jiaotong University, China, 2005. From January 2006 to December 2006, he was an assistant professor in East China Normal University, China. During 2007, he was a research fellow in Nanyang Technological University, Singapore. Now he is with the school of communication and information engineering, Shanghai University, China. His current research interests include image processing, image analysis, and computational intelligence.

Xinpeng Zhang received B.S. degree in computational mathematics from Jilin University, China, in 1995, and M.E. and Ph.D. degrees in communication and information system from Shanghai University, China, in 2001 and 2004, respectively, where he has been with the faculty of the School of Communication and Information Engineering, since 2004, and is currently a professor. His research interests include information hiding, image processing, and digital forensics. He has published over 200 papers in these areas.

Funding
This work was supported by the Natural Science Foundation of China (Grant U1736213, U1636206, and U1936214).

Availability of data and materials
The datasets involved in the current study are available from the corresponding author by reasonable request.

Competing interests
None

Author details
[1]School of Communication and Information Engineering, Shanghai University, Shanghai, China. [2]Shanghai Institute of Intelligent Electronics and Systems, School of Computer Science, Fudan University, Shanghai, China.

Received: 27 March 2020 Accepted: 6 July 2020
Published online: 14 July 2020

References

1. X. Zhang, Reversible data hiding with optimal value transfer. IEEE Trans. Multimedia **15**(2), 316–325 (2012)
2. J. Fridrich, *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge University Press (2009)
3. M. Wu, B. Liu, Data hiding in image and video. I. Fundamental issues and solution. IEEE Trans. Image Process. **12**(6), 685–695 (2003)
4. T. Denemark, V. Sedighi, V. Holub, et al., in *IEEE Int. Workshop on Information Forensics and Security (WIFS)*. Selection-channel-aware rich model for steganalysis of digital images (2014), pp. 48–53
5. T.D. Denemark, M. Boroumand, J. Fridrich, Steganalysis features for content-adaptive JPEG steganography. IEEE Trans. Inf. Forensics Secur. **11**(8), 1736–1746 (2016)
6. Z. Qian, Z. Wang, X. Zhang, et al., Breaking steganography: slight modification with distortion minimization. Int. J. Digit. Crime Forensics **11**(1), 114–125 (2019)
7. P.H. Chen. Adding privacy protection to photo uploading. Tagging in social networks, U.S. Patent 8,744,143, 2014.
8. C. Rey, G. Doërr, G. Csurka, et al., Toward generic image dewatermarking. Proc. IEEE Int. Conf. Image Process. **3**, 633–636 (2002)
9. A.H. Taherinia, M. Jamzad, Blind dewatermarking method based on wavelet transform. Opt. Eng. **50**(5), 057006 (2011)
10. P. Nikbakht, M. Mahdavi, in *IEEE 5th Int. Conf. Comput. Knowl. Eng (ICCKE)*. Targeted dewatermarking of two non-blind SVD-based image watermarking schemes (2015), pp. 80–86
11. J. Tao, S. Li, X. Zhang, et al., Towards robust image steganography. IEEE Trans. Circuits Syst. Video Techno. **29**(2), 594–600 (2018)
12. J. Wu, Y. Yu, C. Huang, et al., in *Proc. IEEE Conf. computer vision and pattern recognition*. Deep multiple instance learning for image classification and auto-annotation (2015), pp. 3460–3469
13. B. Zhu, J.Z. Liu, S.F. Cauley, et al., Image reconstruction by domain-transform manifold learning. Nature **555**(7697), 487 (2018)
14. Y. Sun, D. Liang, X. Wang, et al., Deepid3: face recognition with very deep neural networks. arXiv preprint arXiv:1502.00873 (2015)
15. C. Ledig, L. Theis, F. Huszár, et al., in *Proc. IEEE Conf. computer vision and pattern recognition*. Photo-realistic single image super-resolution using a generative adversarial network (2017), pp. 4681–4690
16. J. Xie, L. Xu, E. Chen, in *Advances in neural information processing systems*. Image denoising and inpainting with deep neural networks (2012), pp. 341–349
17. H.C. Burger, C.J. Schuler, S. Harmeling, in *IEEE Conf. computer vision and pattern recognition*. Image denoising: can plain neural networks compete with BM3D? (2012), pp. 2392–2399
18. X. Zhang, R. Wu, in *IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*. Fast depth image denoising and enhancement using a deep convolutional network (2016), pp. 2499–2503
19. K. Zhang, W. Zuo, Y. Chen, et al., Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising. IEEE Trans. Image Process. **26**(7), 3142–3155 (2017)
20. K. Zhang, W. Zuo, S. Gu, et al., in *Proc. IEEE Conf. computer vision and pattern recognition*. Learning deep CNN denoiser prior for image restoration (2017), pp. 3929–3938
21. I. Goodfellow, J. Pouget-Abadie, M. Mirza, et al., in *Advances in neural information processing systems*. Generative adversarial nets (2014), pp. 2672–2680
22. B. Chen, G.W. Wornell, Quantization index modulation: a class of provably good methods for digital watermarking and information embedding. IEEE Trans. Inf. Theory **47**(4), 1423–1443 (2001)
23. R.C. Dixon, *Spread spectrum systems: with commercial application* (Wiley, New York, 1994)
24. X. Kang, J. Huang, W. Zeng, Efficient general print-scanning resilient data hiding based on uniform log-polar mapping. IEEE Trans. Inf. Forensics Secur. **5**(1), 1–12 (2010)
25. A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv: 1511.06434 (2015)
26. T.Y. Lin, M. Maire, S. Belongie, et al., in *European Conf. computer vision*. Microsoft COCO: common objects in context (Springer, Cham, 2014), pp. 740–755

Publisher's Note