Model Watermarking for Image Processing Networks

Jie Zhang[†],[1] **Dongdong Chen**[†*],[2] **Jing Liao,**[3] **Han Fang,**[1]
Weiming Zhang,[1] **Wenbo Zhou,**[1] **Hao Cui,**[1] **Nenghai Yu**[1]
[1]University of Science and Technology in China [2]Microsoft Cloud AI [3]City University of Hong Kong
[1]{zjzac@mail., fanghan@mail., zhangwm@, welbeckz@mail., cvhc@mail., ynh@}ustc.edu.cn
[2]cddlyf@gmail.com [3]jingliao@cityu.edu.hk

Abstract

Deep learning has achieved tremendous success in numerous industrial applications. As training a good model often needs massive high-quality data and computation resources, the learned models often have significant business values. However, these valuable deep models are exposed to a huge risk of infringements. For example, if the attacker has the full information of one target model including the network structure and weights, the model can be easily finetuned on new datasets. Even if the attacker can only access the output of the target model, he/she can still train another similar surrogate model by generating a large scale of input-output training pairs. How to protect the intellectual property of deep models is a very important but seriously under-researched problem. There are a few recent attempts at classification network protection only.

In this paper, we propose the first model watermarking framework for protecting image processing models. To achieve this goal, we leverage the spatial invisible watermarking mechanism. Specifically, given a black-box target model, a unified and invisible watermark is hidden into its outputs, which can be regarded as a special task-agnostic barrier. In this way, when the attacker trains one surrogate model by using the input-output pairs of the target model, the hidden watermark will be learned and extracted afterward. To enable watermarks from binary bits to high-resolution images, both traditional and deep spatial invisible watermarking mechanism are considered. Experiments demonstrate the robustness of the proposed watermarking mechanism, which can resist surrogate models learned with different network structures and objective functions. Besides deep models, the proposed method is also easy to be extended to protect data and traditional image processing algorithms.

Introduction

In recent years, deep learning has revolutionized a wide variety of tasks such as image recognition (Krizhevsky, Sutskever, and Hinton 2012; He et al. 2016), medical image processing (Hong et al. 2017b; 2019; Zhang et al. 2017a; Hong et al. 2017a), speech recognition (Graves, Mohamed, and Hinton 2013; Zhang et al. 2017b) and natural language processing (Vaswani et al. 2017), and significantly outperforms traditional state-of-the-art methods. To fully utilize the strong learning capability of these deep models and avoid overfitting, a large scale of high-quality labeled data and massive computation resources are often required. Since both human annotation and computation resources are expensive, these learned models are of great business value and need to be protected. But compared to traditional image watermarking techniques, protecting the intellectual property (IP) of deep models is much more challenging. Because of the exponential search space of network structures and weights, numerous structure and weight combinations exist for one specific task. In other words, we can achieve similar or better performance even if we slightly change the structure or weights of the target model.

In the white-box case, where the full information including the detailed network structure and weights of the target model is known, one typical and effective attacking way would be fine-tuning or pruning based on the target model on new task-specific datasets. Even in the black-box case, where only the output of the target model can be accessed, we can still steal the intellectual property of the target model by using another surrogate model to imitate its behavior. Specifically, we can first generate a large scale of input-output training pairs based on the target model then directly train the surrogate model in a supervised manner by regarding the outputs of the target model as ground-truth labels.

Very recently, some research works (Uchida et al. 2017; Adi et al. 2018; Zhang et al. 2018; Merrer, Perez, and Trédan 2017) start paying attention to the IP protection problem for deep neural networks. They often either add a parameter regularizer to the loss function or use the predictions of a special set of indicator images as the watermarks. However, deep watermarking is still a seriously under-researched field and all existing methods only consider the classification task. And in real scenarios, labeling the training data for image processing tasks, is much more complex and expensive than classification tasks, because their ground-truth labels should be pixel-wisely precise. Examples include removing all the ribs in Chest X-ray images and the rain streaks in real rainy images. Therefore protecting such image processing models is more valuable.

[*]Corresponding author, † Equal contribution.

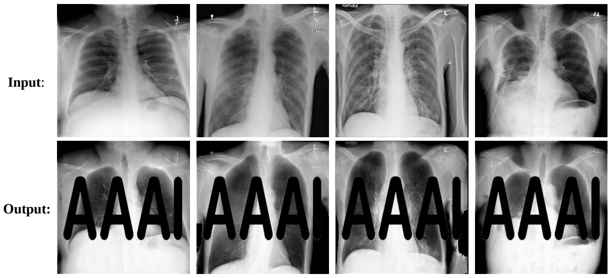

Figure 1: The simplest watermarking mechanism by adding unified visible watermarks onto the target output images, which will sacrifice the visual quality and usability.

Motivated by this, this paper considers the deep watermarking problem for image processing models for the first time. And because the original raw model does not need to be provided in most application scenarios, they can be easily encrypted with traditional algorithms to resist the white-box attack (i.e., fine-tuned or pruned). So we mainly consider the second black-box attack case where only outputs of the target model can be obtained and attackers use surrogate models to imitate it. To resist such attacks, the designed watermarking mechanism should guarantee the watermarks can still be extracted from outputs of learned surrogate models.

Before diving into the model watermarking for image processing networks, we first discuss the simplest spatial visible watermarking mechanism shown in Figure 1. Suppose that we have a lot of input-output training pairs and we manually add a unified visible watermark template to all the outputs. Intuitively, if a surrogate model is trained on such pairs with the simple L2 loss, the learned model will learn this visible watermark into its output to get lower loss. That is to say, given one target model, if we forcibly add one unified visible watermark into all its output, it can resist the plagiarism from other surrogate models to some extent. However, the biggest limitation of this method is that the added visible watermarks will sacrifice the visual quality and usability of the target model seriously. Another potential threat is that attackers may use image editing tools like Photoshops to manually remove all the visible watermarks.

To address the above limitations, we propose a general model watermarking framework by leveraging the spatial invisible watermarking mechanism as shown in Figure 2. Given a target model $\mathbf{M}$ to be protected, we denote its original input and output images as domain $\mathbf{A}$ and $\mathbf{B}$ respectively. Then a spatial invisible watermark embedding method $\mathbf{H}$ is used to hide a unified target watermark $\boldsymbol{\delta}$ into all the output images in the domain $\mathbf{B}$ and generate a new domain $\mathbf{B}'$. Different from the above simple visible watermarks, all the images in the domain $\mathbf{B}'$ should be visually consistent to domain $\mathbf{B}$. Symmetrically, given the images in domain $\mathbf{B}'$, the corresponding watermark extracting algorithm $\mathbf{R}$ will extract the watermark $\boldsymbol{\delta'}$ out, which is consistent to $\boldsymbol{\delta}$. The key hypothesis here is that when the attacker uses $\mathbf{A}$ and $\mathbf{B}'$ to learn a surrogate model $\mathbf{SM}$, $\mathbf{R}$ can still extract the target watermark from the output $\mathbf{B}''$ of $\mathbf{SM}$.

We first test the effectiveness of our framework by using traditional spatial invisible watermarking algorithms like (Kutter 1999; Voloshynovskiy, Deguillaume, and Pun 2000). It works well for some surrogate models but limited to some other ones. Another big limitation is that the information capacity they can hide is relatively low, e.g., tens of bits. To hide high capacity watermarks like logo images and achieve better robustness, we propose a novel deep invisible watermarking system shown in Figure 3, which consists of two main parts: one embedding sub-network $\mathbf{H}$ to learn how to hide invisible watermarks into the image, and another extractor sub-network $\mathbf{R}$ to learn how to extract the invisible watermark out. To avoid $\mathbf{R}$ generating watermark for all the images no matter whether they have invisible watermarks or not, we also constrain it not to extract any watermark out if its input is a clean image. To further boost the robustness, another adversarial training stage is used.

Experiments show that the proposed method can resist the attack from surrogate models trained with different network structures like Resnet and UNet and different loss functions like $L1$, $L2$, perceptual loss and adversarial loss. Depending on the specific task, we find it is also possible to combine the functionality of $\mathbf{M}$ and $\mathbf{H}$ to train a task-specific $\mathbf{H}$.

To summarize, our contributions are fourfold:

- We are the first to introduce the intellectual property protection problem for image processing tasks. We hope it can draw more attention to this research field and inspire greater works.
- We propose the first model watermarking framework to protect image processing networks by leveraging the spatial invisible watermarking mechanism.
- We design a novel deep watermarking algorithm to improve the robustness and capacity of traditional spatial invisible watermarking methods.
- Extensive experiments demonstrate that the proposed framework can resist the attack from surrogate models trained with different network structures and loss functions. It can also be easily extended to protect valuable data and traditional algorithms.

Related work

Media Watermarking Algorithms. Watermarking is one of the most important ways to protect media copyright. For image watermarking, many different algorithms have been proposed in the past decades, which can be roughly categorized into two types: visible watermarks like logos, and invisible watermarks. Compared to visible watermarks, invisible watermarks are more secure and robust. They are often embedded in the original spatial domain (Kutter 1999; Voloshynovskiy, Deguillaume, and Pun 2000; Deguillaume, Voloshynovskiy, and Pun 2002; Voloshynovskiy, Deguillaume, and Pun 2001), or other image transform domains such as discrete cosine transform (DCT) domain (Hsu and Wu 1999; Hernandez, Amado, and Perez-Gonzalez 2000), discrete wavelet transform (DWT) domain (Barni, Bartolini, and Piva 2001), and discrete Fourier transform (DFT) domain (Ruanaidh, Dowling, and Boland 1996). However, all these traditional watermarking algorithms are often only able to hide several or tens of bits, let alone explicit logo images. More importantly, we find only spatial domain watermarking work to some extent for this task and all other transform domain watermarking algorithms fail.

In recent years, some DNN-based watermarking schemes have been proposed. For example, Zhu *et al.*(Zhu et al. 2018) propose an auto-encoder-based network architecture to realize the embedding and extracting process of watermarks. Based on it, Tancik *et al.*(Tancik, Mildenhall, and Ng 2019) further realize a camera shooting resilient watermarking scheme by adding a simulated camera shooting distortion to the noise layer. Compared to these image watermarking algorithms, model watermarking is much more challenging because of the exponential search space of deep models. But we innovatively find it possible to leverage spatial invisible watermarking techniques for model protection.

Model Watermarking Algorithms. Though watermarking for deep neural networks are still seriously under-studied, there are some recent works (Uchida et al. 2017; Adi et al. 2018; Nagai et al. 2018; Zhang et al. 2018) that start paying attention to it. For example, based on the over-parameterized property of deep neural networks, Uchida *et al.*(Uchida et al. 2017) propose a special weight regularizer to the objective function so that the distribution of weights can be resilient to attacks such as fine-tuning and pruning. One big limitation of this method is not task-agnostic and need to know the original network structure and parameters for retraining. Adi *et al.*(Adi et al. 2018) use a particular set of inputs as the indicators and let the model deliberately output specific incorrect labels, however, it may not work if the network are retrained. Zhang *et al.*(Zhang et al. 2018) associate the watermark with the actual identity by making great changes to original images, which is easy to be detected.

However, all the methods mentioned above focus on the classification tasks, which is different from the purpose of this paper: protecting higher commercial valued image processing models. We innovatively leverage spatial invisible watermarking algorithms for image processing networks and propose a new deep invisible watermarking technique to enable high-capacity watermarks(e.g., logo images).

Image-to-image Translation Networks. In the deep learning era, most image processing tasks such as image segmentation, edge to the image, deraining, and X-ray image debone, can be solved with an image-to-image translation network where the input and output are both images. Recently this field has achieved significant progress especially after the emergence of the generative adversarial network (GAN) (Goodfellow et al. 2014). Isola *et al.*propose a general image-to-image translation framework by combining adversarial training in (Isola et al. 2017), which is further improved by many following works (Choi et al. 2018; Wang et al. 2018; Park et al. 2019). The limitation of these methods is that they need a lot of pairwise training data. By introducing the cycle consistency, Zhu *et al.*propose a general unpaired image-to-image translation framework CycleGAN (Zhu et al. 2017). In this paper, we mainly focus on the deep models of paired image-to-image translation, because the paired training data is much more expensive to be obtained than classification datasets or unpaired datasets. More importantly, there is no prior work that has ever considered the watermarking issue for such models.

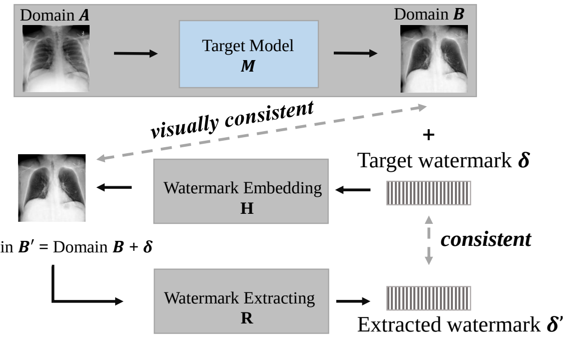

Figure 2: The proposed deep watermarking framework by leveraging spatial invisible watermarking algorithms.

Method

In this section, the details will be elaborated. Before that, we will first introduce the formal problem definition and give a simple theoretical pre-analysis to justify our hypothesis.

Problem Definition. For image processing tasks, assume the input domain $\mathbf{A}$ is composed of massive images $\{a_1, a_2, ..., a_n\}$, and the target output domain $\mathbf{B}$ consists of images $\{b_1, b_2, ..., b_n\}$. In this paper, we only consider the pairwise case where a_i and b_i are one-one matched by an implicit transformation function $\mathcal{T}$. Then the goal of the image processing model $\mathbf{M}$ is to approximate $\mathcal{T}$ by minimizing the distance $\mathcal{L}$ between $\mathbf{M}(a_i)$ and b_i, i.e.,

$$\mathcal{L}(\mathbf{M}(a_i), b_i) \to 0. \tag{1}$$

Assume we have learned a target model $\mathbf{M}$ based on massive private image pairs and computation resources. Given each input image a_i in domain $\mathbf{A}$, $\mathbf{M}$ will output an image b_i in domain $\mathbf{B}$. Then the attacker may use the image pairs defined by (a_i, b_i) from domain $\mathbf{A}, \mathbf{B}$ to train another surrogate model $\mathbf{SM}$. Our target is to design an effective deep watermarking mechanism which is able to identify $\mathbf{SM}$ once it is trained with data generated by $\mathbf{M}$. Because in real scenarios, it is highly possible we cannot access the raw model $\mathbf{SM}$ in a white-box way, the only indicator we can leverage is the output of $\mathbf{SM}$. Therefore, we need to figure out one way to extract watermarks from the output of $\mathbf{SM}$.

Theoretical Pre-analysis. In traditional watermarking algorithms, given an image I and a target watermark $\boldsymbol{\delta}$ to embed, they will first use a watermark embedding algorithm $\mathbf{H}$ to generate an image I' which contains $\boldsymbol{\delta}$. Symmetrically, the target watermark $\boldsymbol{\delta}$ can be further extracted out with the corresponding watermarking extracting algorithm $\mathbf{R}$. Considering each image $b_i \in \mathbf{B}$ is embedded with a unified watermark $\boldsymbol{\delta}$, where $b'_i = b_i + \boldsymbol{\delta}$, forming another domain $\mathbf{B}'$, there must exist a model $\mathbf{M}'$ which can learn good transformation between domain $\mathbf{A}$ and $\mathbf{B}'$. One simplest solution of $\mathbf{M}'$ is to directly add $\boldsymbol{\delta}$ to the output of $\mathbf{M}$ with a skip connection:

$$\begin{aligned} &\mathcal{L}(\mathbf{M}(a_i), b_i) \to 0 \Leftrightarrow \mathcal{L}(\mathbf{M}'(a_i), (b_i + \delta)) \to 0 \\ &\text{when} \quad \mathbf{M}' = \mathbf{M}(a_i) + \delta. \end{aligned} \tag{2}$$

Based on the above observation, we propose a general deep watermarking framework for image processing models shown in Figure 2. Given a target model $\mathbf{M}$ to protect, we add a barrier $\mathbf{H}$ by embedding a unified watermark $\boldsymbol{\delta}$ into all its output images before showing them to the end-users. So the surrogate model $\mathbf{SM}$ has to be trained with the image pair (a_i, b_i') from domain $\mathbf{A}, \mathbf{B}'$ with watermark, instead of the original pair (a_i, b_i) from domain $\mathbf{A}, \mathbf{B}$. No matter what architecture $\mathbf{SM}$ adopts, it's behavior will approach to that of $\mathbf{M}'$ in preserving the unified watermark $\boldsymbol{\delta}$. Otherwise, its objective loss function $\mathcal{L}$ cannot achieve a lower value. And then the watermarking extracting algorithm $\mathbf{R}$ can extract the watermark from the output of $\mathbf{SM}$.

To ensure the watermarked output image b_i' is visually consistent with the original one b_i, only spatial invisible watermarking algorithms are considered in this paper. Below we will try both traditional spatial invisible watermarking algorithm and a novel deep invisible watermarking algorithm.

Traditional Spatial Invisible Watermarking. Additive-based embedding is the most common method used in the traditional spatial invisible watermarking scheme. The watermark is first spread to a sequence or block which satisfies a certain distribution, then embedded into the corresponding coefficients of the host image. This embedding procedure can be formulated by

$$I' = \begin{cases} I + \alpha C_0 & \text{if } w_i = 0 \\ I + \alpha C_1 & \text{otherwise} \end{cases} \tag{3}$$

where I and I' indicate the original image and embedded image respectively. α indicates the embedding intensity and C_i denote the spread image block that represents bit "w_i"$(w_i \in [0, 1])$. In the extraction side, the watermark is determined by detecting the distribution of the corresponding coefficients. The robustness of such an algorithm is guaranteed by the spread spectrum operation. The redundancy brought by the spread spectrum makes a strong error correction ability of the watermark so that the distribution of the block will not change a lot even after image processing.

However, such algorithms often have very limited embedding capacity because many extra redundant bits are needed to ensure robustness. In fact, in many application scenarios, the IP owners may want to embed some special images (e.g., logos) explicitly, which is nearly infeasible for these algorithms. More importantly, the following experiments show that these traditional algorithms can only resist some special types of surrogate models. To enable more high-capacity watermarks and more robust resistance ability, we propose a new deep invisible watermarking algorithm and utilize a two-stage training strategy as shown in Figure 3.

Deep Invisible Watermarking. To embed an image watermark into host images of the domain $\mathbf{B}$ and extract it out afterward, one embedding sub-network $\mathbf{H}$ and one extractor sub-network $\mathbf{R}$ are adopted respectively. Without sacrificing the original image quality of domain $\mathbf{B}$, we require images with the hidden watermark should be still visually consistent with the original images in the domain $\mathbf{B}$. Since adversarial networks have demonstrated their power in reducing the domain gap in many different tasks, we append one discriminator network $\mathbf{D}$ after $\mathbf{H}$ to further improve the image quality of domain $\mathbf{B}'$. During training, we find if the extractor network $\mathbf{R}$ is only trained with the images of domain $\mathbf{B}'$, it is very easy to overfit and output the target watermark no matter whether the input images contain watermarks or not. To avoid it, we also feed the images of domain $\mathbf{A}$ and domain $\mathbf{B}$ that do not contain watermarks into $\mathbf{R}$ and force it to output a constant blank image. In this way, $\mathbf{R}$ will have the real ability to extract watermarks only when the input image has the watermark in it.

Based on the pre-analysis, when the attacker uses a surrogate model $\mathbf{SM}$ to imitate the target model $\mathbf{M}$ based on the input domain $\mathbf{A}$ and watermarked domain $\mathbf{B}'$, $\mathbf{SM}$ will learn the hidden watermark $\boldsymbol{\delta}$ into its output thanks to the inherent fitting property of deep networks. Despite of higher hiding capacity, similar to traditional watermarking algorithms, the extractor sub-network $\mathbf{R}$ cannot extract the watermarks out from the output of the surrogate model $\mathbf{SM}$ neither if only with this initial training stage. This is because $\mathbf{R}$ has only observed clean watermarked images but not the watermarked images from surrogate models which may contain some unpleasant noises. To further enhance the extracting ability of $\mathbf{R}$, we choose one simple surrogate network to imitate the attackers' behavior and fine-tune $\mathbf{R}$ on the mixed dataset of domain $\mathbf{A}, \mathbf{B}, \mathbf{B}', \mathbf{B}''$. Experiments show this will significantly boost the extracting ability of $\mathbf{R}$ and resist other types of surrogate models.

Network Structures. In our method, we adopt the UNet (Ronneberger, Fischer, and Brox 2015) as the default network structure of $\mathbf{H}$ and $\mathbf{SM}$, which has been widely used by many translation based tasks like (Isola et al. 2017; Zhu et al. 2017). It performs especially well for tasks where the output image shares some common properties of input image by multi-scale skip connections. But for the extractor sub-network $\mathbf{R}$ whose output is different from the input, we find CEILNet (Fan et al. 2017) works much better. It also follows an auto-encoder like network structure. In details, the encoder consists of three convolutional layers, and the decoder consists of one deconvolutional layer and two convolutional layers symmetrically. To enhance the learning capacity, nine residual blocks are inserted between the encoder and decoder. For the discriminator $\mathbf{D}$, we adopt the PatchGAN (Isola et al. 2017) by default. Note that except for the extractor sub-network, we find other types of translation networks also work well in our framework, which demonstrates the strong generalization ability of our framework.

Loss Functions. The objective loss function of our method consists of two parts: the embedding loss $\mathcal{L}_{emd}$ and the extracting loss $\mathcal{L}_{ext}$, i.e.,

$$\mathcal{L} = \mathcal{L}_{emd} + \lambda * \mathcal{L}_{ext}, \tag{4}$$

where λ is the hyper parameter to balance these two loss terms. Below we will introduce the detailed formulation of $\mathcal{L}_{emd}$ and $\mathcal{L}_{ext}$ respectively.

Embedding Loss. To embed the watermark image while guaranteeing the original visual quality, three different types of visual consistency loss are considered: the basic $L2$ loss ℓ_{bs}, perceptual loss ℓ_{vgg}, and adversarial loss ℓ_{adv}, i.e.,

$$\mathcal{L}_{emd} = \lambda_1 * \ell_{bs} + \lambda_2 * \ell_{vgg} + \lambda_3 * \ell_{adv}. \tag{5}$$

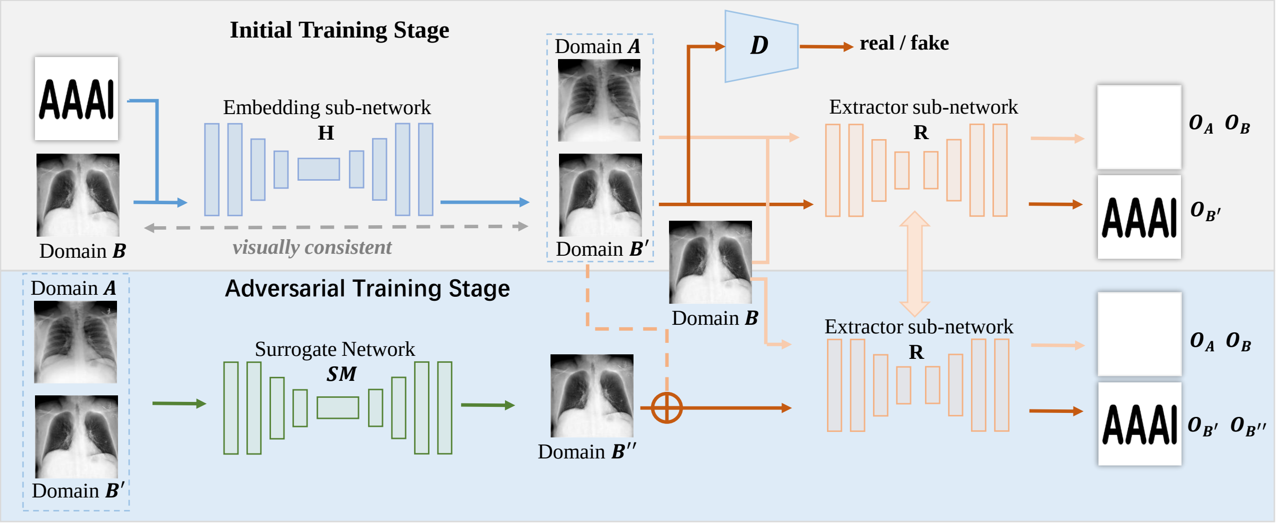

Figure 3: The overall pipeline of the proposed deep invisible watermarking algorithm and two-stage training strategy. In the first training stage, a basic watermark embedding sub-network $\mathbf{H}$ and extractor sub-network $\mathbf{R}$ are trained. Then another surrogate network $\mathbf{SM}$ is leveraged as the adversarial competitor to further enhance the extracting ability of $\mathbf{R}$.

Here the basic $L2$ loss ℓ_{bs} is simply the pixel value difference between the input host image b_i and the watermarked output image b'_i, N_c is the total pixel number, i.e.,

$$\ell_{bs} = \sum_{b'_i \in \mathbf{B}', b_i \in \mathbf{B}} \frac{1}{N_c} \|b'_i - b_i\|^2. \tag{6}$$

And the perceptual loss ℓ_{vgg} (Johnson, Alahi, and Fei-Fei 2016) is defined as the difference between the VGG feature of b_i and b'_i:

$$\ell_{vgg} = \sum_{b'_i \in \mathbf{B}', b_i \in \mathbf{B}} \frac{1}{N_f} \|VGG_k(b'_i) - VGG_k(b_i)\|^2, \tag{7}$$

where $VGG_k(\cdot)$ denotes the features extracted at layer k ("conv2_2" by default), and N_f denotes the total feature neuron number. To further improve the visual quality and minimize the domain gap between $\mathbf{B}'$ and $\mathbf{B}$, the adversarial loss ℓ_{adv} will let the embedding sub-network $\mathbf{H}$ hide watermarks better so that the discriminator $\mathbf{D}$ cannot differentiate its output from real watermark-free images in $\mathbf{B}$, i.e.,

$$\ell_{adv} = \mathop{\mathbb{E}}_{b_i \in \mathbf{B}} log(\mathbf{D}(b_i)) + \mathop{\mathbb{E}}_{b'_i \in \mathbf{B}'} log(1 - \mathbf{D}(b'_i)). \tag{8}$$

Extracting Loss. The responsibility of the extractor sub-network R has two aspects: it should be able to extract the target watermark out for watermarked images from $\mathbf{B}', \mathbf{B}''$ and output a constant blank image for watermark-free images from $\mathbf{A}, \mathbf{B}$. So the first two terms of $\mathcal{L}_{ext}$ are the reconstruction loss ℓ_{wm} and ℓ_{clean} for these two types of images respectively, i.e.,

$$\begin{aligned} \ell_{wm} &= \sum_{b'_i \in \mathbf{B}'} \frac{1}{N_c} \|R(b'_i) - \delta\|^2 + \sum_{b''_i \in \mathbf{B}''} \frac{1}{N_c} \|R(b''_i) - \delta\|^2, \\ \ell_{clean} &= \sum_{a_i \in \mathbf{A}} \frac{1}{N_c} \|R(a_i) - \delta_0\|^2 + \sum_{b_i \in \mathbf{B}} \frac{1}{N_c} \|R(b_i) - \delta_0\|^2, \end{aligned} \tag{9}$$

where δ_0 is the constant blank watermark image. Besides reconstruction loss, we also want the watermarks extracted from different watermarked images to be consistent, thus another consistent loss is added:

$$\ell_{cst} = \sum_{x,y \in \mathbf{B}' \cup \mathbf{B}''} \|R(x) - R(y)\|^2. \tag{10}$$

Then $\mathcal{L}_{ext}$ is defined as the weighted sum of these three terms, i.e.,

$$\mathcal{L}_{ext} = \lambda_4 * \ell_{wm} + \lambda_5 * \ell_{clean} + \lambda_6 * \ell_{cst}. \tag{11}$$

Adversarial Training Stage. With the above initial training stage, $\mathbf{R}$ only observes the clean watermarked images and cannot generalize well for the noisy watermarked output of some surrogate models. To enhance its extracting ability, an extra adversarial training stage is added. Specifically, one surrogate model $\mathbf{SM}$ is trained with the simple $L2$ loss by default. Denote the outputs of $\mathbf{SM}$ as $\mathbf{B}''$, we further finetune $\mathbf{R}$ on the mixed dataset $\mathbf{A}, \mathbf{B}, \mathbf{B}', \mathbf{B}''$ in this stage.

Experiments

In this paper, two examples of image processing tasks are conducted: image deraining and Chest X-ray image debone. The goal of these two tasks is to remove the rain streak and rib components from the input images respectively. To demonstrate the effectiveness of our method, we first show the newly introduced deep invisible watermarking algorithm can hide high-capacity image-based watermarks, then evaluate the robustness of the proposed deep watermarking framework to different surrogate models. Finally, some ablation analysis is provided to justify the motivation of our design and shed some light on more inspiring potentials.

Implementation Details. For image deraining, we use 12100 images from the PASCAL VOC dataset as target domain $\mathbf{B}$, and use the synthesis algorithm in (Zhang and Pa-

Task	PSNR	SSIM	NC
Debone-aaai	47.89	0.99	0.9999
Derain-flower	39.98	0.99	0.9966

Table 1: Quantitative results of the proposed invisible image based watermarking. *-aaai and *-flower use the AAAI logo and a colorful flower image as watermarks respectively.

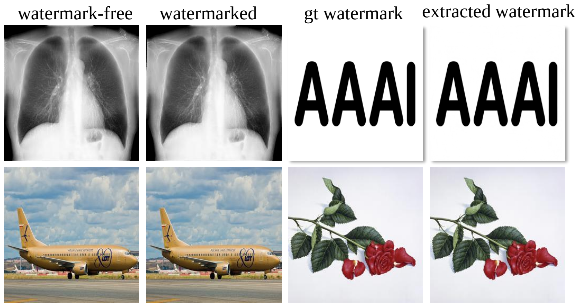

Figure 4: Two examples of hiding image watermark into host images with the proposed invisible watermarking algorithm.

tel 2018) to generate rainy images as domain $\mathbf{A}$. These images are split into three parts: 6000 both for the initial and adversarial training, 6000 to train the surrogate model and 100 for testing. Similarly, for X-ray image debone, we select 6100 high-quality chest X-ray images from the open dataset chestx-ray8 (Wang et al. 2017) and use the rib suppression algorithm proposed by (Yang et al. 2017) to generate the training pair. They are also divided into three parts: 3000 both for the initial and adversarial training, 3000 to train the surrogate model and 100 for testing. By default, $\lambda, \lambda_1, \lambda_2, \lambda_4, \lambda_5, \lambda_6$ all equal to 1 and $\lambda_3 = 0.01$.

Evaluation Metric. To evaluate the visual quality, PSNR and SSIM are used by default. To judge whether the watermark is extracted successfully, we use the classic normalized correlation (NC) metric as previous watermarking methods. The watermark is regarded as successfully extracted if its NC value is bigger than 0.95. Based on it, the success rate (SR) is further defined as the ratio of watermarked images whose hidden watermark is successfully extracted.

Deep Image Based Invisible Watermarking. In this experiment, we give both quantitative and qualitative results about the proposed deep image based invisible watermarking algorithm. For debone and deraining, one AAAI logo image and a colorful flower image are used as the example watermark image respectively. As shown Table 1, the embedding sub-network $\mathbf{H}$ and the extractor sub-network $\mathbf{R}$ collaborate very well. $\mathbf{H}$ can hide the image watermark into the host images invisibly with high visual quality (average PSNR 39.98 and 47.89 for derain and debone respectively), and $\mathbf{R}$ can extract these hidden watermarks out afterward with an average NC value over 0.99 (100% success rate). Two visual examples are shown in Figure 4.

Robustness to The Attack from Surrogate Models. To evaluate the final robustness of the proposed deep watermarking framework, we use a lot of surrogate models that are trained with different network structures and objective loss functions to imitate the attackers' behavior. Here four different types of network structures are considered: vanilla convolutional networks only consisting of several convolutional layers ("CNet"), an auto-encoder like networks with 9 and 16 residual blocks ("Res9", "Res16"), and the aforementioned UNet network ("UNet"). For objective loss functions, some popular loss functions like $L1$, $L2$, perceptual loss L_{perc}, adversarial loss L_{adv} and their combination are considered. Since one surrogate model with "UNet" and $L2$ loss function is leveraged in the adversarial training stage, this configuration can be viewed as a white-box attack and all other configurations are black-box attacks.

Setting	T-Debone	T-Derain	D-Debone	D-Derain	D-Debone†	D-Derain†
CNet	0%	0%	92%	100%	0%	0%
Res9	0%	0%	100%	100%	0%	0%
Res16	0%	0%	100%	100%	0%	0%
UNet	100%	100%	100%	100%	0%	0%

Table 2: The success rate (SR) of resisting the attack from surrogate models trained with $L2$ loss but different network structures. T-* means the results of using traditional spatial invisible watermarking algorithms to hide 64-bit, while D-* means that of the proposed deep invisible watermarking algorithm to hide watermark images. † denotes the results without adversarial training.

Due to the limited computation resource, we do not consider all the combinations of different network structures and loss functions. Instead, we choose to conduct the control experiments to demonstrate the robustness to the network structures and loss functions respectively. In the Table 2, both traditional spatial bit-based invisible watermarking algorithms (hide 64-bit) and the proposed deep image-based invisible watermarking algorithm are tested. Though only UNet based surrogate model trained with $L2$ loss is leveraged in the adversarial training stage, we find the proposed deep model watermarking framework can resist both white-box and black-box attacks when equipped with the newly proposed deep image-based invisible watermarking technique. For traditional watermarking algorithms, they can only resist the attack of some special surrogate models because their extracting algorithms cannot handle noisy watermarked images from different surrogate models. More importantly, they cannot hide high-capacity watermarks like logo images. We have also tried many traditional *transform domain watermarking algorithms* like DCT-based(Fang et al. 2018),DFT-based(Kang, Huang, and Zeng 2010) and DWT-based(Kang et al. 2003) *but all of them do not work and achieve* 0% *success rate.*

To further demonstrate the robustness to different losses, we use the UNet as the default network structure and train surrogate models with different combinations of loss functions. As shown in Table 3, the proposed deep watermarking framework has a very strong generalization ability and can resist different loss combinations with very high success rate. Since in real user scenarios, detailed network structure and training objective functions are the parts of the surrogate model that attacker may often change, we have enough rea-

Task	$L1$	$L1 + L_{adv}$	$L2$	$L2 + L_{adv}$	L_{perc}	L_{perc}+L_{adv}
D-Debone	100%	100%	100%	100%	88%	92%
D-Derain	100%	100%	100%	100%	86%	100%
D-Debone†	0%	98%	0%	100%	0%	0%
D-Derain†	0%	0%	0%	100%	24%	0%

Table 3: The success rate (SR) of resisting the attack from surrogate models that are trained with different loss combinations. † means the results without adversarial training.

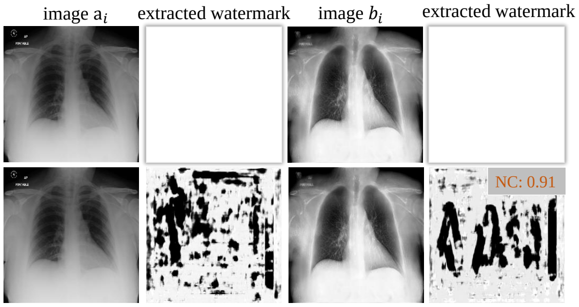

Figure 5: Comparison results with (first row) and without (second row) clean loss. The second and last column are the extracted watermarks from the watermark-free images a_i, b_i from domain $\mathbf{A}, \mathbf{B}$ respectively.

sons to think the proposed deep watermarking framework is applicable in these cases.

Ablation Study

The Importance of Clean Loss and Consistent Loss. Besides the watermark reconstruction loss, we add one clean loss and consistent loss into the extracting loss. To demonstrate their importance, two control experiments are conducted. As shown in Figure 5, without clean loss, the extractor will always extract meaningless watermark from watermark-free images of domain $\mathbf{A}, \mathbf{B}$. Especially for images of domain $\mathbf{B}$, the extracted watermarks have a quite large NC value and make the forensics meaningless. Similarly in Figure 6, we find the extractor can only extract very weak watermarks or even cannot extract any watermark out when training without consistent loss. By contrast, our method can always extract very clear watermarks out.

The Importance of Adversarial Training. As described above, to enhance the extracting ability of $\mathbf{R}$, another adversarial training stage is used. To demonstrate its necessity, we also conduct the control experiments without adversarial training, and attach the corresponding results in Table 2 and Table 3 (labelled with †). It can be seen that, with the default $L2$ loss, its resisting success rate is all about 0% for surrogate models of different network structures. When using UNet as the network structure but training with different losses, we find only some special surrogate models can partially extract the hidden watermarks, which demonstrates the significant importance of the adversarial training.

Task Specific Deep Invisible Watermarks. In our default setting, the embedding and extractor sub-network are task-agnostic and appended as a general barrier. In this experiment, we try a more challenging task that lets $\mathbf{H}$ be task-specific. Take debone as the example, the input image of $\mathbf{H}$ is directly the image of the domain $\mathbf{A}$ with rib components now, and we want $\mathbf{H}$ to remove the rib components and hide watermarks simultaneously. In such a case, $\mathbf{H}$ is the final watermarked target model $\mathbf{M}$. For comparison, we also train a baseline debone model without the need of hiding watermarks. We find the above task-specific watermarked model can achieve very comparable results (PSNR: 24.49, SSIM: 0.91) to this baseline model (PSNR: 25.81, SSIM: 0.91).

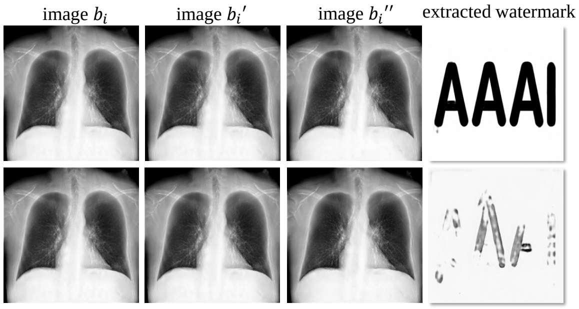

Figure 6: Comparison results with (first row) and without (second row) consistent loss. The last column is the extracted watermark from the output b_i'' of surrogate model.

Extension to Protect Data and Traditional Algorithms. Though our motivation is to protect deep models, the proposed framework is easy to be extended to protect valuable data or traditional algorithms by directly embedding the watermarks into their labeled groundtruth images or outputs.

Conclusion

We introduce the deep watermarking problem for image processing networks for the first time.Inspired by traditional spatial invisible media watermarking, the first deep watermarking framework is proposed. To make it robust to different surrogate models and support image-based watermarks, we propose a novel deep invisible watermarking technique. Experiments demonstrate that our framework can resist the attack from surrogate models trained with different network structures and loss functions. We hope this work can inspire more great works for this seriously under-researched field.

Acknowledgments

This work was partially supported by the Natural Science Foundation of China under Grant U1636201, 61572452.

References

Adi, Y.; Baum, C.; Cisse, M.; Pinkas, B.; and Keshet, J. 2018. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *USENIX*.

Barni, M.; Bartolini, F.; and Piva, A. 2001. Improved wavelet-based watermarking through pixel-wise masking. *TIP* 10(5):783–791.

Choi, Y.; Choi, M.; Kim, M.; Ha, J.-W.; Kim, S.; and Choo,

J. 2018. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*.

Deguillaume, F.; Voloshynovskiy, S. V.; and Pun, T. 2002. Method for the estimation and recovering from general affine transforms in digital watermarking applications. In *SWMC*, volume 4675, 313–322.

Fan, Q.; Yang, J.; Hua, G.; Chen, B.; and Wipf, D. 2017. A generic deep architecture for single image reflection removal and image smoothing. In *ICCV*, 3238–3247.

Fang, H.; Zhang, W.; Zhou, H.; Cui, H.; and Yu, N. 2018. Screen-shooting resilient watermarking. *TIFS*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NIPS*, 2672–2680.

Graves, A.; Mohamed, A.-r.; and Hinton, G. 2013. Speech recognition with deep recurrent neural networks. In *ICASSP*, 6645–6649. IEEE.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Hernandez, J. R.; Amado, M.; and Perez-Gonzalez, F. 2000. Dct-domain watermarking techniques for still images: Detector performance analysis and a new structure. *TIP*.

Hong, S.; Wu, M.; Li, H.; and Wu, Z. 2017a. Event2vec: Learning representations of events on temporal sequences. In *APWEB-WAIM*, 33–47. Springer.

Hong, S.; Wu, M.; Zhou, Y.; Wang, Q.; Shang, J.; Li, H.; and Xie, J. 2017b. Encase: An ensemble classifier for ecg classification using expert features and deep neural networks. In *CinC*. IEEE.

Hong, S.; Zhou, Y.; Wu, M.; Shang, J.; Wang, Q.; Li, H.; and Xie, J. 2019. Combining deep neural networks and engineered features for cardiac arrhythmia detection from ecg recordings. *PMEA*.

Hsu, C.-T., and Wu, J.-L. 1999. Hidden digital watermarks in images. *TIP* 8(1):58–68.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. *CVPR*.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 694–711. Springer.

Kang, X.; Huang, J.; Shi, Y. Q.; and Lin, Y. 2003. A dwt-dft composite watermarking scheme robust to both affine transform and jpeg compression. *TCSVT* 13(8):776–786.

Kang, X.; Huang, J.; and Zeng, W. 2010. Efficient general print-scanning resilient data hiding based on uniform log-polar mapping. *TIFS* 5(1):1–12.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*, 1097–1105.

Kutter, M. 1999. Watermarking resistance to translation, rotation, and scaling. In *MSA*, volume 3528, 423–431.

Merrer, E. L.; Perez, P.; and Trédan, G. 2017. Adversarial frontier stitching for remote neural network watermarking. *arXiv*.

Nagai, Y.; Uchida, Y.; Sakazawa, S.; and Satoh, S. 2018. Digital watermarking for deep neural networks. *IJMIR*.

Park, T.; Liu, M.-Y.; Wang, T.-C.; and Zhu, J.-Y. 2019. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2337–2346.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 234–241. Springer.

Ruanaidh, J.; Dowling, W.; and Boland, F. M. 1996. Phase watermarking of digital images. In *ICIP*. IEEE.

Tancik, M.; Mildenhall, B.; and Ng, R. 2019. Stegastamp: Invisible hyperlinks in physical photographs. *arXiv*.

Uchida, Y.; Nagai, Y.; Sakazawa, S.; and Satoh, S. 2017. Embedding watermarks into deep neural networks. In *ICMR*, 269–277. ACM.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS*, 5998–6008.

Voloshynovskiy, S.; Deguillaume, F.; and Pun, T. 2000. Content adaptive watermarking based on a stochastic multiresolution image modeling. In *EUSIPCO*, 1–4. IEEE.

Voloshynovskiy, S.; Deguillaume, F.; and Pun, T. 2001. Multibit digital watermarking robust against local nonlinear geometrical distortions. In *ICIP*, volume 3, 999–1002.

Wang, X.; Peng, Y.; Lu, L.; Lu, Z.; Bagheri, M.; and Summers, R. M. 2017. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *CVPR*.

Wang, T.-C.; Liu, M.-Y.; Zhu, J.-Y.; Tao, A.; Kautz, J.; and Catanzaro, B. 2018. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*.

Yang, W.; Chen, Y.; Liu, Y.; Zhong, L.; Qin, G.; Lu, Z.; Feng, Q.; and Chen, W. 2017. Cascade of multi-scale convolutional neural networks for bone suppression of chest radiographs in gradient domain. *Medical image analysis* 35.

Zhang, H., and Patel, V. M. 2018. Density-aware single image de-raining using a multi-stream dense network. In *CVPR*, 695–704.

Zhang, J.; Chen, Y.; Hong, S.; and Li, H. 2017a. Rebuild: graph embedding based method for user social role identity on mobile communication network. In *DMBD*. Springer.

Zhang, Y.; Pezeshki, M.; Brakel, P.; Zhang, S.; Bengio, C. L. Y.; and Courville, A. 2017b. Towards end-to-end speech recognition with deep convolutional neural networks. *arXiv*.

Zhang, J.; Gu, Z.; Jang, J.; Wu, H.; Stoecklin, M. P.; Huang, H.; and Molloy, I. 2018. Protecting intellectual property of deep neural networks with watermarking. In *ASIACCS*, 159–172. ACM.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2223–2232.

Zhu, J.; Kaplan, R.; Johnson, J.; and Fei-Fei, L. 2018. Hidden: Hiding data with deep networks. In *ECCV*, 657–672.