Robust Model Watermarking for Image Processing Networks via Structure Consistency

Jie Zhang*, Dongdong Chen*,†, Jing Liao, Zehua Ma, Han Fang, Weiming Zhang, Huamin Feng, Gang Hua, *Fellow, IEEE*, and Nenghai Yu

Abstract—The intellectual property of deep networks can be easily "stolen" by surrogate model attack. There has been significant progress in protecting the model IP in classification tasks. However, little attention has been devoted to the protection of image processing models. By utilizing consistent invisible spatial watermarks, the work [1] first considered model watermarking for deep image processing networks and demonstrated its efficacy in many downstream tasks. Its success depends on the hypothesis that if a consistent watermark exists in all prediction outputs, that watermark will be learned into the attacker's surrogate model. However, when the attacker uses common data augmentation attacks (e.g., rotate, crop, and resize) during surrogate model training, it will fail because the underlying watermark consistency is destroyed. To mitigate this issue, we propose a new watermarking methodology, "structure consistency", based on which a new deep structure-aligned model watermarking algorithm is designed. Specifically, the embedded watermarks are designed to be aligned with physically consistent image structures, such as edges or semantic regions. Experiments demonstrate that our method is more robust than the baseline in resisting data augmentation attacks. Besides that, we test the generalization ability and robustness of our method to a broader range of adaptive attacks.

Index Terms—Deep Model IP Protection, Model Watermarking, Image Processing

1 Introduction

DEEP learning has made tremendous success in many application domains, including computer vision [2], [3], natural language processing [4], and autonomous driving [5], to name a few. However, it is often not that easy to train a good DNN model because of the demand for massive training data and computation resources. Recently, for business consideration, protecting the intellectual property (IP) of DNN models has attracted much attention from both academia and industry. However, it is still a seriously under-explored field because of its inherent challenges.

The challenges indeed come from the powerful learning capacity of DNN, which is a double-edged sword. On the one hand, it makes discriminative feature representation learning easy in different tasks once sufficient high-quality data is granted. On the other hand, the attacker can use one surrogate model to imitate one target network's behavior even if the network structure and weights are both unknown. For example, with the model API at the cloud platform, the attacker can first feed a lot of inputs into the API and obtain their outputs. The attacker then regards such input-output pairs as training samples and distills a good surrogate model, similar to the teacher-student learning. This attack is called "surrogate model attack" or "model extraction attack" [6], [7].

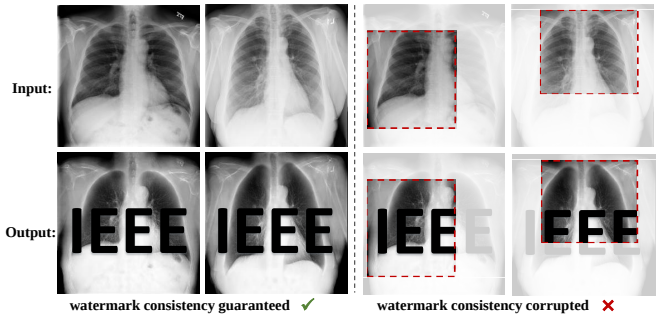

Fig. 1. Left: the working principle of [1], which embeds a unified (consistent) watermark into network outputs to guarantee the assumed watermark consistency; Right: its fragility to regular augmentation techniques like random cropping, which will destroy the watermark consistency.

In order to protect the model IP, many methods [8], [9] have been proposed. However, most of them focus on the classification task and only consider modification-based attacks like "fine-tuning" and "network pruning", where the attacker has access to the victim model, including its parameters and architectures. Recently, the work [1] began to consider the IP protection problem for image processing networks and surrogate model attacks. The motivation of this work is very straightforward. As shown in the left part of Figure 1, they embed a unified watermark (e.g., same embed position, watermark size, etc.) into the target model output. When the attacker learns a surrogate model by using the input-output pairs from the target model, the surrogate model will also learn this unified watermark into its outputs to minimize the training loss. Considering their watermarks are essentially a unified watermark image in different embedded outputs, we regard it as "whole-image consistency".

Notwithstanding its success, the "whole-image consistency"

- *Jie Zhang, Zehua Ma, Han Fang, Weiming Zhang, and Nenghai Yu are with the School of Cyber Science and Security, University of Science and Technology of China, Hefei, Anhui 230026, China. E-mail: {zjzac@mail., mzh045@mail., fanghan@mail., zhangwm@, ynh@}ustc.edu.cn. This work is partially finished when Jie Zhang is a research fellow at Nanyang Technological University, Singapore.*
- *Dongdong Chen is with Microsoft Research, Redmond, Washington 98052, USA, E-mail: cddlyf@gmail.com*
- *Jing Liao is with the Department of Computer Science, City University of Hong, E-mail: jingliao@cityu.edu.hk*
- *Huaming Feng is with Beijing Electronic Science and Technology Institute, E-mail: fenghm@besti.edu.cn*
- *Gang Hua is with Wormpex AI Research LLC, WA 98004, US, E-mail: ganghua@gmail.com*
- *∗ Equal contributions. † Dongdong Chen is the corresponding author.*

can be easily destroyed by common augmentation techniques such as random cropping and rotation, which is explained as a big limitation in its extension work [10]. The reason is illustrated in the right part of Figure 1. In this augmented case, the surrogate model cannot find a consistent watermark pattern, thus directly ignoring it as random noise by considering all the training samples.

To address the above limitation, we propose a new watermarking methodology in this paper, which is inherently robust to data augmentation. Rather than pursuing the above whole-image consistency, we design "structure consistency", which couples the watermark patterns with the image structures. It is inspired by the fact that some global structures such as edges or local semantic structures such as eyes can keep their physical meaning after augmentation. If we embed the watermark information into such structures, the watermark consistency can be naturally preserved. Based on this observation, we design a structure-aligned watermarking scheme, which encodes the watermark information into constant color values and fills them into the above structure regions as a special type of watermark.

The overall model watermarking framework is shown in Figure 3. It basically consists of four modules: a watermark bit encoder to encode watermark bits into structure-aligned watermarks, an embedding network that learns to embed structure-aligned watermarks into the cover images without sacrificing the original visual quality, an extracting network that tries to extract hidden watermarks out from watermarked images, and a final decoder to decode the recovered bits. For the integrity of forensics, the extracting network will output a blank image for all unwatermarked images. However, training such a framework to achieve great performance is not a trivial task, because the hidden watermark information will be easily destroyed under diverse augmentations. To overcome this issue, we further design an incremental training strategy, which adds new augmentation operators or loss constraints gradually. Extensive experiments demonstrate the superior performance and robustness of our method. Our contributions are four-fold:

- We provide detailed analysis regarding the fragility of the watermarking scheme proposed in [1], [10], and explain why the whole-image consistency is not robust to data augmentation.
- We propose a new methodology called "structure consistency", based on which a structure-aligned model watermarking framework is designed.
- To circumvent the learning difficulty, an incremental training strategy is designed by gradually involving new augmentation operators or loss constraints.
- We demonstrate the superior robustness of our method in different application scenarios and adaptive attacks.

2 Related Work

Model Watermarking. Model watermarking (*i.e*. DNN watermarking) is a popular technique for model IP protection. In recent years, for the classification task, several algorithms [8], [9] have been proposed. In [8], a special weight regularizer is leveraged so that the distribution of model weights can be represented as watermarks. However, it only works in a white-box way, which needs to know the original network structure and parameters. To remedy it, some black-box DNN watermarking methods are proposed, most of which are designed based on backdoor attacks. For example, Adi *et al.* [9] uses a particular set of inputs as the indicators and lets the model deliberately output specific incorrect labels. Despite their success, most of these methods only concentrate on simple modification-based attacks like fine-tuning and model pruning. In [11], [12], [13], [14], [15], [16], some works started to consider the more challenging surrogate model attack but still only focus on the classification task.

Recently, the work of [1] starts to consider the watermarking problem for image processing networks and innovatively leveraged spatial invisible watermarking algorithms for model watermarking against surrogate model attack. However, as pointed out in their extension work [10], it will totally fail when the attacker utilizes some data augmentation during the surrogate model's training, as the underlying working principle relies on the whole-image watermark consistency. Concurrently, Wu *et al*. [17] also try to protect the IP of image processing network, but without considering the surrogate model attack. Besides, Wu's method is also very similar to the case called as "self-watermarked models" in the work of [10]. Therefore, we regard the work of [1] as the baseline for comparison, and our method is motivated by the baseline, but designs the new structure consistency to obtain augmentation robustness.

Data Augmentation. Data augmentation plays a crucial role in learning better and generalized deep neural networks. Basic data augmentation techniques include rotation, flipping, cropping, and adding noises. Depending on whether they affect the original image quality, we divide them into two categories: quality-harmless and quality-harmful. For deep image processing tasks, since the attacker wants the surrogate model to get high-quality output, we mainly consider the common quality-harmless data augmentation techniques such as flipping, rotation, cropping, and resizing. Besides, we also consider 6 quality-harmful augmentations (namely, noise, blur, hue, saturation, contrast and style transfer) as an ablation study to test the robustness.

Image-to-Image Translation. Image-to-image translation is a typical image processing task of which the input and output are both images. It is widely adopted in many applications such as edge to the image synthesis, deraining, and X-ray Chest image debone. In recent years, Generative Adversarial Network (GAN) [3] has brought significant progress to image-to-image translation. Generally, there are three typical settings: paired [18], [19], unpaired [20], and semi-paired [21]. Similar as [1], [10], because paired data is more difficult and expensive to collect and many high-quality deep processing models are also trained in a supervised way, we mainly consider the pairwise translation as the example applications.

3 Pre-analysis and Motivation

Recap of the "whole-image consistency". Given an input domain $A = \{a_1, a_2, ..., a_n\}$ and a target output domain $B = \{b_1, b_2, ..., b_n\}$, pairwised deep image processing is to learn a good target model $\mathbf{M}$ so that $\mathbf{M}(a_i)$ can approach b_i under some pre-defined distance metric $\mathcal{L}$:

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

For surrogate model attack, it means that given a target $\mathbf{M}$, the attacker does not know its detailed network structure and weights but can access $\mathbf{M}$ to get a lot of input-output pairs. Because attacker may use an input set different from that used by $\mathbf{M}$, we denote the generated input-output pairs as $\{a'_1, a'_2, ..., a'_m\}$ and $\{b'_1, b'_2, ..., b'_m\}$ respectively. Then the attacker will use such pairs

Fig. 2. Two repetitive watermark patterns tried to preserve the whole-image consistency in the baseline method. But still no watermark can be extracted from the surrogate model's outputs.

to train a surrogate model $\mathbf{SM}$. The working principle of [1], [10] is based on the hypothesis that if $\mathbf{SM}$ can learn a good mapping between $\{a'_1, a'_2, ..., a'_m\}$ and $\{b'_1, b'_2, ..., b'_m\}$, then if a unified watermark δ is added to all the output b'_i, $\mathbf{SM}$ will also absorb δ into its output, which can be extracted out for forensics. This is based on the fact that:

$$\mathcal{L}(\mathbf{SM}(a'_i), b'_i) \to 0 \Leftrightarrow \mathcal{L}(\mathbf{SM}'(a'_i), b'_i + \delta) \to 0 \quad \text{when} \quad \mathbf{SM}' = \mathbf{SM} + \delta. \tag{2}$$

Because of the fitting and loss minimization property of deep networks, $\mathbf{SM}$ can be easily learned to be $\mathbf{SM}'$ by adding a skip connection δ. Since δ is a unified watermark image embedded in different outputs, we call it "whole-image consistency".

Fragility to Data Augmentation. Despite the effectiveness, it has a serious limitation as admitted in [10], *i.e.* , the above "whole-image consistency" is not robust to data augmentation techniques, which are commonly used in training DNNs. Because data augmentation will destroy the underlying watermark consistency, which is the basis of [1], [10]. To preserve the consistency, one intuitive way is to use repetitive watermark patterns as shown in Figure 2, and train the framework with different augmentation operators. However, we find it still does not work. Briefly, no watermark pattern can be extracted from the surrogate model's outputs, and the successful extracting rate (SR) is 0% in all cases. Because even though the watermark pattern is repetitive, it will still change during the augmentation process, *e.g.* position shift during cropping and orientation change during rotation.

In fact, we find that the whole-image consistency is indeed methodologically difficult to hold under data augmentation. Specifically, denote the data augmentation operation of each $\{a'_i, b'_i + \delta\}$ as T_i, the surrogate model $\mathbf{SM}$ trained with augmentation is to learn the mapping between the domain $\{T_1(a'_1), T_2(a'_2), ..., T_m(a'_m)\}$ and $\{T_1(b'_1 + \delta), T_2(b'_2 + \delta), ..., T_m(b'_m + \delta)\}$. Let us simplify the explanation by assuming T_i to be a linear operation, *i.e.* , $T_m(b'_m + \delta) = T_m(b'_m) + T_m(\delta)$. For pair-wised image processing, since there exists underlying content relationship between a'_i and b'_i, if the same constant T_0 ($T_i = T_0$) is applied to all the (a'_i, b'_i), once the target model $\mathbf{M}$ can learn such a mapping relationship, it should be feasible for $\mathbf{SM}$. However, ***if different*** T_i ***is used for different*** i***, and*** δ ***is not content-related to*** a'_i ***or*** b'_i***, then*** $T_i(\delta)$ ***will lose its consistency across different*** i ***and is not related to*** a'_i, b'_i ***either***. In this case, $\mathbf{SM}$ is impossible to learn δ into its output anymore. This is because, without the consistency constraint or content relationship, given $T_i(a_i)$, there is no information available for $\mathbf{SM}$ to predict what $T_i(\delta)$ looks like, thus $\mathbf{SM}$ directly regards it as independent noise and ignore it by considering the whole training set.

Structure Consistency. As analyzed above, if we want $\mathbf{SM}$ to absorb the watermark δ, δ must be able to keep its consistency under data augmentation. A trivial solution is to let δ be a pure-color image with constant pixel values. However, such a pure-color watermark image is unfriendly to the convolutional watermark extracting network. Because if the extracting network needs to extract such a constant δ out for different watermarked images, the convolutional weights will be learned to all zeros while only bias term being non-zeros. In this way, even given an unwatermarked image, it will also output δ too, which loses the forensics meaning. Therefore, we resort to a more advanced way: making the watermark pattern consistent with image structures.

It is inspired by the observation that some global structures like edges or some local semantic structures like "eyes" of the face are content-related and can keep their physical meaning under the common data augmentation techniques, we call this type of consistency "structure consistency". By further encoding the watermark information into specific color values and filling them into these consistent structures, we can generate structure-aligned watermark δ_i for each a'_i, b'_i. During the augmentation T_i, δ_i will adaptively change along with a'_i, b'_i and keep its alignment with structures of a'_i, b'_i. Therefore, it is still possible for $\mathbf{SM}$ to absorb δ_i based on such structure consistency.

4 Structure-aligned Model Watermarking

Overview. Based on the above structure consistency analysis, we propose the structure-aligned model watermarking algorithm in Figure 3. The basic goal is to learn a good embedding network $HNet$ and a corresponding extracting network $EXNet$. $HNet$ is responsible to embed the structure-aligned watermarks into the cover images to generate watermarked images while $EXNet$ is responsible to extract the embedded watermarks out. To make $HNet$ and $EXNet$ robust to different augmentation operations, an augmentation layer is inserted between them and jointly trained. After the training, given a target model to protect, we feed its output to $HNet$ before exposing it to the public. In such a way, the outputs obtained by the attackers are watermarked. And if one surrogate model is trained with such input and watermarked output pairs, $EXNet$ can still extract the target watermarks from the surrogate model's outputs for forensics. Besides, one bit encoder and decoder are leveraged to encode/decode the watermark bits respectively. Below we will elaborate on each part in detail. For ease of presentation, we will use b_i below as the substitute for b'_i.

Watermark Bit Encoder and Structure Extractor. We propose to fill constant RGB values into these structures to ensure consistency in the physical structures during the augmentation process. Taking the common 8-bit color space as an example, the value range of each color channel ("Red", "Blue," and "Green") would be $[0, 255]$. Assuming the color step used for encoding is t, then the total number of possible pixel values n equals $\frac{255}{t} \cdot \frac{255}{t} \cdot \frac{255}{t}$ (255 left as unwatermarked indicator). Therefore, the max watermark bit sequence length is $\lfloor log_2(n) \rfloor$. In real applications, the watermark bit sequence $\mathcal{S}$ represents the IP information we want to embed, such as company name, model ID, and version.

Given the $\mathcal{S}$, we can use some simple mathematical encoding schemes (eg., hash functions) to map $\mathcal{S}$ into one specific color value $\mathcal{C}_i$. The detailed physical structure format may be different depending on the specific task. For example, we can use the global edges for general natural images and local semantic regions like "eyes" or "noses" for face images. In the following experiments, we will try three different types of physical structures to demonstrate the generality of our method. By default, we use the well-

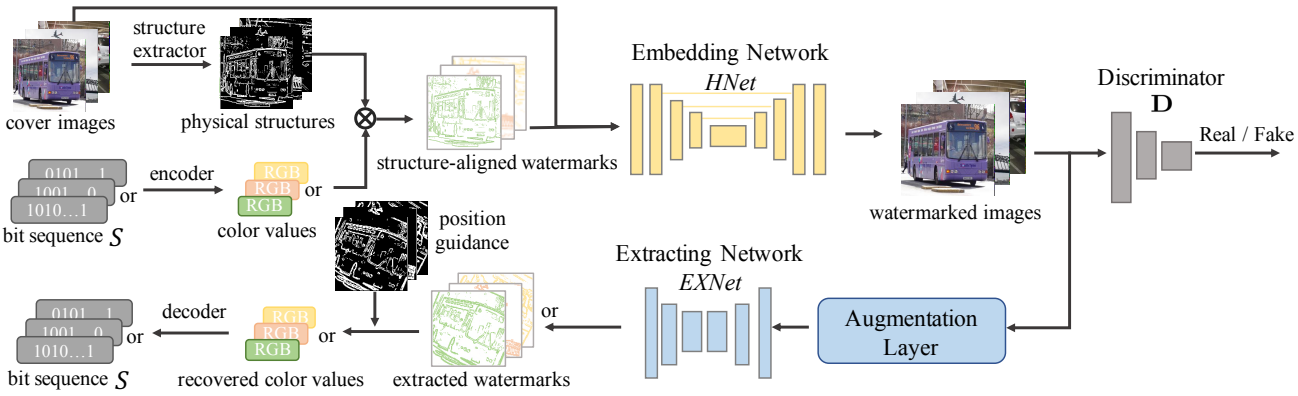

Fig. 3. The proposed structure-aligned model watermarking framework, which consists of four modules: watermark bit encoder and decoder, one embedding network $HNet$ and an extracting network $EXNet$, where the augmentation layer is inserted between $HNet$ and $EXNet$. For better performance, a discriminator network $\mathbf{D}$ is also appended.

known Sobel edge algorithm to extract the global edges as the physical structure.

Structure-aligned Model Watermarking. After getting the encoded color $\mathcal{C}_i$ and structure map $\mathcal{M}_i$, we fill $\mathcal{M}_i$ with $\mathcal{C}_i$ to produce a structure-aligned watermark $\mathcal{W}_i$:

$$\mathcal{W}_i = \mathcal{C}_i \otimes \mathcal{M}_i. \tag{3}$$

Here, $\otimes$ means filling $\mathcal{C}_i$ into the regions of $\mathcal{M}_i$ whose mask values are 1 and filling a blank color (R:255,G:255,B:255) otherwise. As we want $HNet$ be capable of handling different $\mathcal{C}_i$ rather than use an independent $HNet$ for each possible $\mathcal{C}_i$, we randomly sample different C_i during training.

After obtaining $\mathcal{W}_i$, we concatenate it with the original cover image b_i along the channel dimension and feed them into $HNet$ to get the watermarked image b_i^w. To ensure the robustness to different augmentation operators $\{T_1, ..., T_k\}$, b_i^w will be randomly processed by one or multiple augmentation operators before being fed into the extracting network $EXNet$. Then, we will recover the hidden color values from the extracted watermark $\mathcal{W}_i'$ by using the physical structure of the watermarked image as the position guidance. Finally, the original watermark bit sequence will be decoded from the recovered color values.

Network Structures. For fair comparison, we follow [1] and adopt the UNet [22] as our $HNet$. It is an auto-encoder like network structure and adds multiple skip connections between the encoder and decoder part, which is a widely used design in many image translation tasks. For the extracting network $EXNet$, we also adopt an auto-encoder like network structure. Specifically, three convolutional layers are used as the encoder and one deconvolutional layer along with two convolutional layers are regarded as the decoder. Several residual blocks are further inserted between the encoder and decoder to enhance its learning capacity. To help achieve better visual quality, we leverage one patch discriminator network $\mathbf{D}$ [18] for adversarial training.

Loss Functions. The training loss consists of two parts: the embedding loss $\mathcal{L}_H$ and the extracting loss $\mathcal{L}_{EX}$:

$$\mathcal{L} = \mathcal{L}_H + \lambda \cdot \mathcal{L}_{EX}, \tag{4}$$

where λ is the hyper parameter to balance their importance. $\mathcal{L}_H$ is to ensure the visual quality of watermarked images while $\mathcal{L}_{EX}$ is to ensure that the hidden watermarks can be successfully extracted out. Therefore, a too large λ will cause inferior visual quality but higher extracting success rate, and too small λ will obtain high visual quality watermarked images but the hidden watermark would be too weak to be extracted out.

The embedding loss $\mathcal{L}_H$ has two parts: a simple $L2$ loss ℓ_2 and an adversarial loss ℓ_{adv}, i.e.,

$$\mathcal{L}_H = \lambda_1 \cdot \ell_2 + \lambda_2 \cdot \ell_{adv}. \tag{5}$$

The $L2$ loss ℓ_2 measures the pixel-wise difference between the input cover image b_i and the watermarked output image b_i^w. That is to say, we want the watermarked images to be visually similar to the original unwatermarked images so that the attacker even cannot know whether the output of the target model is watermarked or not:

$$\ell_2 = \mathop{\mathbb{E}}_{b_i \in \mathbf{B}, b_i^w \in \mathbf{B}^w} \|b_i - b_i^w\|^2, \tag{6}$$

here, $\mathbf{B}$ and $\mathbf{B}^w$ represent the unwatermarked and watermarked image set respectively. And the adversarial loss ℓ_{adv} will encourage the embedding network $HNet$ to hide watermarks better so that the discriminator $\mathbf{D}$ cannot distinguish its output from real unwatermarked images b_i,

$$\ell_{adv} = \mathop{\mathbb{E}}_{b_i \in \mathbf{B}} log(\mathbf{D}(b_i)) + \mathop{\mathbb{E}}_{b_i^w \in \mathbf{B}^w} log(1 - \mathbf{D}(b_i^w)). \tag{7}$$

For effective forensics, besides the requirement that $EXNet$ can extract the hidden watermarks out from the watermarked images, we also need $EXNet$ not to extract any watermark out for unwatermarked images. Therefore, the extracting loss consists of two terms: one for watermarked images ℓ_{wm} and one for unwatermarked images:

$$\mathcal{L}_{EX} = \lambda_3 \cdot \ell_{wm} + \lambda_4 \cdot \mathop{\mathbb{E}}_{b_i \in \mathbf{B}} \|EXNet(b_i) - \mathcal{O}\|^2, \tag{8}$$

where $\mathcal{O}$ represents the constant image with all pixels values as (R:255,G:255,B:255) for unwatermarked images. To balance the loss contributions from the watermarked and unwatermarked regions, an adaptive weight λ_5 will be added for watermarked regions. Formally, ℓ_{wm} is defined as:

$$\begin{aligned} \ell_{wm} = & \lambda_5 \cdot \mathop{\mathbb{E}}_{b_i^w \in \mathbf{B}^w} \|EXNet(b_i^w) \otimes \mathcal{M}_i - \mathcal{W}_i\|^2 \\ & + \mathop{\mathbb{E}}_{b_i^w \in \mathbf{B}^w} \|EXNet(b_i^w) \otimes \overline{\mathcal{M}_i} - \mathcal{O}\|^2. \end{aligned} \tag{9}$$

As defined before, $\mathcal{M}_i$ represents the physical structure region, $\overline{\mathcal{M}_i}$ is the background region and $\mathcal{W}_i$ denotes the ground-truth watermark. The weight λ_5 depends on the ratio of the physical structure area to the total image area. The smaller the ratio, the larger

the weight λ_5. In our implementation, λ_5 is pre-calculated on a set of training images to ensure $\lambda_5 \cdot \sum_i \|\mathcal{M}_i\|_1 \approx \sum_i \|\overline{\mathcal{M}_i}\|_1$.

To enhance the ability of $EXNet$ in extracting watermarks from the surrogate models' output, we also add an adversarial training stage as [1]. Specifically, one simple surrogate network is used to mimic the attacker's behavior, then $EXNet$ is fine-tuned by adding outputs of this surrogate model into its training set. This stage can be regarded as one special augmentation operation from model processing.

Incremental Training Strategy. Unlike [1] where the watermarked images are assumed unchanged, watermarked images in our case will be processed under different types of data augmentation. And some augmentation operations will significantly change the original statistics of hidden watermarks and make it more difficult to be extracted. To resist different augmentation operations, we add these augmentation operators into the training process, forming an augmentation layer.

We find training such a system with all operators together from scratch is challenging. To reduce the learning difficulty, we propose an incremental training strategy by adding augmentation operators one by one into training until the previous one converges. For the objective loss function, we only use the ℓ_2 loss term in $\mathcal{L}_H$ to constrain the $HNet$ until all the augmentation operators are added, and then add the adversarial loss ℓ_{adv} to fine-tune the $HNet$ for achieving better visual quality. In the ablation study, effectiveness of this training strategy will be studied. In addition, we clarify the relationship to the baseline method [1] in the supplementary material.

5 Experiments

5.1 Experiment Settings

The proposed method can be broadly used in many different commercial systems for IP protection, such as medical image processing and remote sensing image enhancement. Due to the lack of large public datasets, we tried the two example image processing tasks (deraining, X-ray Chest image debone) used in [1] and a new artistic portrait generation (APG) task to demonstrate our effectiveness. The details of datasets, hyper-parameters and augmentation setting are provided in the appendix. Because of the resource and space consideration, we mainly use the derain task for comparison and ablation. For comparison, as the baseline method [1] is the only effective method to date and other traditional watermark types have already been proved ineffective in [1], [10], we only compare our method with [1].

Recovering Color Values. Given an extracted watermark, we directly use a straightforward algorithm to recover the hidden color value: extracting the physical structure of watermarked images as position guidance and calculating the average value in each color channel as the color value.

Evaluation Metric. PSNR and SSIM are used as the default visual quality metric. For extracting performance, we define the biggest recovered color value error of different color channels as the actual error value and set 10 as the absolute error value threshold ($|\mathrm{TH}|$=10), namely, t=20 and $\lfloor log_2((\frac{255}{20})^3)\rfloor$=11 bits are embedded. When the error value falls in the range of the threshold, we define it as a successful extraction. The successful extracting rate (SR) is the ratio of images with successful extraction. Due to the watermarking mechanism difference, we still use the NC value introduced in [1] to measure the baseline method. Compared to the NC value, our metric is *more strict*.

5.2 Comparison Experiments

Results of Watermarked Images and Extracted Images. To ensure the watermark embedding network $HNet$ can embed the structure-aligned watermarks into the cover image b_i and guarantee the watermarked image b_i^w is visually similar to the b_i, we first evaluate the PSNR and SSIM values between the watermarked images and the original clean images on the test dataset. Results show that our method can obtain visually indistinguishable watermarked images with the PSNR value as 37.86 and the SSIM value as 0.97. Although there exists a slight performance degradation compared with the baseline method (PSNR 37.86 vs 39.98; SSIM 0.97 vs 0.99), it's acceptable because of obtaining the desired robustness. One example visual result is presented in the first row of Figure 4. It can be seen that our method can extract the hidden watermarks out for both unaugmented and augmented images while guaranteeing high visual quality for watermarked images. Need to note that the end users can only see b_i^w but not b_i.

Robustness to Different Types of Surrogate Model Attacks. For fair comparison with [1], we follow its setting and evaluate the robustness to surrogate model attack by using different surrogate models with respect to network structures and loss functions. Specifically, four different network structures are used: 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 function, L_1, L_2, perceptual loss L_{perc}, adversarial loss L_{adv}, and their combinations are adopted. By default, $\mathbf{SM}$ model with "UNet" and L_2 loss is leveraged in the adversarial training stage, therefore this configuration can be viewed as white-box attack and all other configurations are black-box attacks.

For the computation resource consideration, we follow [1] and conduct controlled experiments to demonstrate the robustness to the network structures and loss functions respectively. Specifically, for the comparison regarding different network structure, the $\mathbf{SM}$ model is only trained with $L2$ loss. And for the loss function comparison, the $\mathbf{SM}$ model adopts the UNet by default. Below, we consider two different training settings: without data augmentation like [1] and with data augmentation.

Without data augmentation. In this ideal setting, the attacker does not pre-process the collected input-output pairs. As shown in Table 1, both our method and the baseline [1] are robust to different surrogate networks and loss functions with the adversarial training stage. But without the adversarial stage, our method can still obtain pretty good results for most cases while the baseline [1] almost fails. In this sense, our structure consistency is more robust than the whole-image consistency. For the result L_{perc} without data augmentation, if the verification metric is looser by increasing $|\mathrm{TH}|$ to 20 (also stricter than the NC metric used by the baseline), our method has a higher SR (98%) than the baseline (86%).

With data augmentation. In this more realistic attack scenario, the attacker will utilize data augmentation operators to train the surrogate model $\mathbf{SM}$. As shown in Table 1, our method succeeds in most scenarios after adversarial training while the baseline method [1] totally fails even integrated with the augmentation layer and after adversarial training (marked with ‡), no matter what kinds of network structure or loss function were used. Specifically, we significantly boost the success rate from 0% to above 90% in most cases (indeed **"significant improvement"**). We also observe that the extra adversarial training stage is very

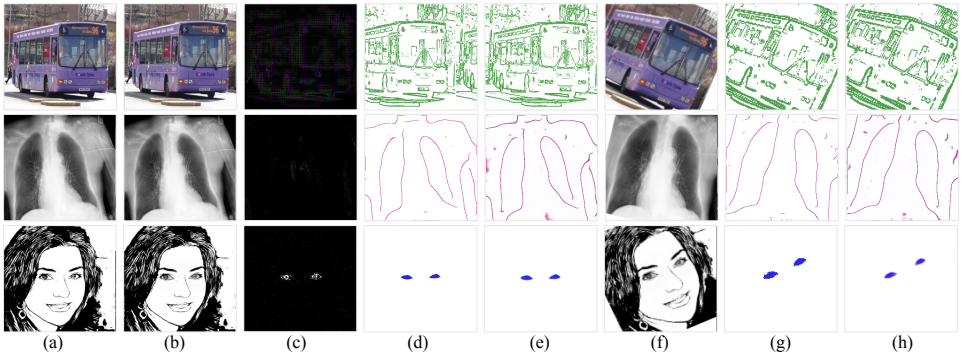

Fig. 4. Some visual results: (a) clean image b_i, (b) watermarked image b_i^w, (c) 10 $\times$ residual between b_i^w and b_i, (d) ground truth watermark $\mathcal{W}_i$, (e) recovered watermark $\mathcal{W}_i'$, (f) $T_i(b_i^w)$ augmented from b_i^w, (g) $T_i(\mathcal{W}_i)$ augmented from $\mathcal{W}_i$, (h) recovered watermark from $T_i(b_i^w)$.

TABLE 1
The success rate of resisting surrogate model attack for different network structures and different loss functions without / with data augmentation (DA). † denotes without the adversarial training stage, ‡ denotes that we integrate the augmentation layer into the original framework of the baseline method [1] , and the false positive rate is 0 for all cases.

DA	Method	Different Network Structures				Different Loss Functions					
		CNet	Res9	Res16	UNet	$L1$	$L1$ + L_{adv}	$L2$	$L2$ + L_{adv}	L_{perc}	L_{perc}+L_{adv}
W/O	[1]	100%	100%	100%	100%	100%	100%	100%	100%	86%	100%
	Ours	100%	100%	100%	100%	100%	100%	100%	100%	59%	100%
	[1]†	0%	0%	0%	0%	0%	0%	0%	100%	24%	0%
	Ours †	84%	82%	84%	45%	54%	99%	45%	99%	0%	98%
W/	[1]	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	[1] ‡	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	Ours	98%	97%	95%	99%	99%	97%	99%	96%	57%	97%
	[1]†	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	Ours †	0%	0%	0%	0%	0%	2%	0%	1%	0%	1%

TABLE 2
The complexity comparison with the baseline [1].

Method	# Model parameter	Training time	Inference time
The baseline	59327619	33h	0.007s
Ours	59327619	81h	0.007s

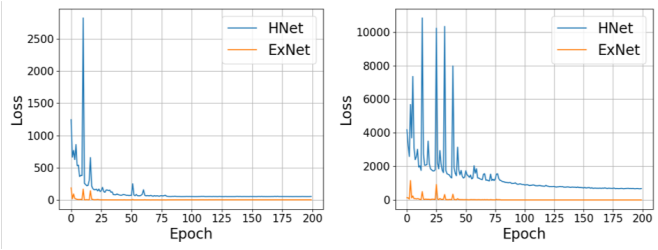

Fig. 5. Loss curve of the baseline (Left) and ours (Right).

TABLE 3
The influence of the hyper-parameter λ.

λ	0.1	1	10
PSNR/SSIM	41.46 / 0.99	37.86 / 0.97	26.85 / 0.84
SR (|TH|=10)	87%	100%	100%

TABLE 4
The influence of the hyper-parameter |TH|.

|TH|	5	10	15
Watermark Bits	14 bits	11 bits	9 bits
SR	96%	100%	100%

important in such challenging data augmentation cases. In the first two rows of Figure 6, we provide some visual results about the extracted watermarks from the outputs of the learned surrogate model. Obviously, after data augmentation, the surrogate model of the baseline cannot learn the watermark into its outputs anymore.

Complexity Comparison with the Baseline. As mentioned above, we adopt the same architecture as the baseline [1] for embedding and extracting. Therefore, as shown in Table 2, the model size (59.3M) and inference time (0.007s for 256×256 input) are the same. Because of the incremental training strategy, the training time of our method is longer than the baseline, e.g., 81h vs 33h on the derain task. In Figure 5, we also show the loss curve of both methods, but it is not an apple-to-apple comparison due to different loss designs.

5.3 Ablation Study

The Influence of Hyper-parameter and Augmentation Setting. We conduct a simple ablation study by using different values for λ. As shown in Table 3 ("|TH|" denotes the absolute error value threshold), a small λ will produce higher visual quality but lower SR values. Therefore, we choose $\lambda = 1$ by default. Besides, lower |TH| means that more watermark bits can be embedded. We further show the SR under different values of |TH| in Table 4. As expected, a decrease in the SR is observed when more watermark bits are embedded. For the influence of the augmentation setting, the experimental results show that all the augmentations are important for robustness. For example, the average SR rate is 78% if resizing is not used, lower than 100% when all the augmentations are used. Besides the default augmentation order in the incremental learning mentioned above, we also tried other order combinations and they also work well (100% SR rate for all cases).

Importance of Incremental Training Strategy. As mentioned above, it is very difficult to train the framework with all the augmentation operations and losses from scratch simultaneously. Therefore, an incremental training strategy is adopted. To justify

Fig. 6. Some extracted outputs of $EXNet$ for different types of inputs: (a) input domain A image a_i, (b) target domain clean image b_i, (c) watermarked image b_i^w, (d) the output of surrogate model. (e)∼(h) are the corresponding extracted results for (a)∼(d). The 1st row is for the baseline [1].

Fig. 7. Watermarked images b_i^w comparison with (middle) and without (right) incremental training strategy. And clean imagesb_i is shown in the 1st column.

its necessity and superiority, we try to train the framework just from scratch rather than training incrementally and show the two watermarked images b_i^w in Figure 7. Obviously, this from-scratch setting suffers from serious color drifting problems. By comparison, it works very well with the proposed incremental training strategy.

5.4 Generalization Ability.

Besides the deraining task, we further apply our framework to the X-ray Chest image debone task, which is also mentioned in [1]. We choose another famous ***Canny*** edge algorithm to extract the global edges as the physical structure. We also try another interesting image processing task, called artistic portrait generation (APG). Given a real face image, APG converts it to a pencil drawing style. To demonstrate the generalization ability of structure consistency, we regard the ***semantic*** "eyes" region as the physical structure. Then the extracting network needs to recognize this semantic structure and extract the hidden watermark out automatically, which is more challenging than global edges based physical structures.

Visual Quality. In the last two rows in Figure 4, we also provide visual examples for debone task and APG task, respectively. For quantitative evaluation, we further calculate the PSNR and SSIM values between the watermarked images and the original clean images on the corresponding test dataset. The PSNR/SSIM values for debone and APG are 44.99 / 0.99 and 37.73 / 0.99 respectively.

Robustness to Surrogate Model Attacks. Here, we only consider the more challenging case, namely, surrogate model attacks with data augmentation. As shown in Table 5, it can also achieve very high extraction robustness on both debone task and APG task in most cases. Note that, when training the surrogate model with L_{perc}, the surrogate model itself produces bad APG results, so the SR is not good in these cases. More experimental results are presented in the supplementary material. In the last two rows of Figure 6, we provide some visual results of both applications. It can be seen that our framework works very well for different physical structures and is general for different tasks.

5.5 Additional Robustness

Robustness to Other Augmentation Attacks. As mentioned before, we only consider quality-harmless augmentation by default and assume all the training pairs used by the surrogate model are the output of our target model. But like the arms race, the attacker may train the surrogate model with partial quality-harmful augmented data or self-labeled data to destroy the consistency constraint and remove the watermark. To simulate such behaviors, we mix some watermarked data augmented by 6 representative quality-harmful techniques and some unwatermarked data into the surrogate model training dataset, respectively. In detail, we add the Gaussian Noise (δ = 0.09) and blur the image with Gaussian Filter (5×5). For color-based augmentations, the shifting range of saturation and contrast are both [0.5, 1.5], and the hue change range is [-0.5, 0.5]. The implementation of style transfer is based on the open-source code [1] of [23]. Specifically, we adopt the style of "starry night". In Table 6, two mixing ratios (10%/50%) are considered. Surprisingly, though the consistency constraint is destroyed in the newly introduced data, our method can still work very well in resisting surrogate model attacks, even when 50% self-labeled clean data is added. Note that we do not retrain our framework here. More visual results and more robustness analysis against augmentation attacks can be found in the supplementary material.

Robustness to More Adaptive Attacks. Apart from data augmentation attacks, attackers may consider more strategies to remove the model watermark adaptively. First, we consider Neural Cleanse [24], which is famous for reverse-engineering the watermark pattern. But it totally fails because our method is designed in a global

1. https://github.com/eriklindernoren/Fast-Neural-Style-Transfer

TABLE 5
The success rate of resisting surrogate model attack for different loss functions with data augmentation on the APG task. † denotes the results without the adversarial training stage.

Task	Different Network Structures				Different Loss Functions					
	CNet	Res9	Res16	UNet	$L1$	$L1 + L_{adv}$	$L2$	$L2 + L_{adv}$	L_{perc}	L_{perc}+L_{adv}
debone	94%	92%	96%	100%	100%	81%	100%	88%	71%	62%
APG	95%	98%	100%	100%	100%	99%	100%	93%	29%	13%

TABLE 6
The image quality and successful extracting rate of our framework for surrogate models trained by mixing some augmented data from other augmentation techniques or clean data. A / B represents the results with 10% and 50% mixing ratios, respectively. Without any augmentation, PSNR and SR are 32.02 and 100%, respectively.

Settings	Noise	Blurring	Hue	Saturation	Contrast	Style Transfer	Clean
PSNR	31.67 / 31.29	31.99 / 31.89	32.02 / 32.03	31.99 / 31.93	32.00 / 31.97	31.76 / 31.06	32.05 / 32.67
SR	100% / 100%	100% / 100%	100% / 98%	100% / 100%	100% / 100%	99% / 99%	100% / 68%

and structure-aligned way, which does not fulfill its assumption, *i.e.* , *the watermark is input-agnostic and static both in location and pattern*. Second, we assume the attacker collects a small amount of clean (unwatermarked) data pair, and conducts supervised fine-tuning. Results show that even fine-tuning with a new same-size clean data, our $EXNet$ can still work well with 78% success rate. Moreover, we also conduct model pruning on the stolen surrogate model. Different from classification models, the performance of the surrogate model degrades a lot (PSNR: from 32.02 to 28.41) when only 30% parameters are pruned. In such case, we only obtain a 6% success rate. However, if we adjust the absolute error value threshold (TH) from default 10 to 25, we can obtain a desirable success rate (86%) again. Third, we consider the case where the attacker has un-paired clean data and trains the surrogate model with a domain-adversarial loss (watermarked vs. non-watermarked images). In this case, the extracting success rate degrades to 43% but it is still acceptable. Moreover, doing this will hurt the surrogate model's performance (PSNR: from 32.02 to 28.9) and make the attack less meaningful. Finally, we consider the robustness of our method to watermark overwriting. Similar to traditional media watermarking, overwriting can be solved by watermark legal agreement. On the other hand, after overwriting, our method can still extract the original watermark out, and the surrogate model performance will degrade a lot, which is similar to the baseline [1], [10].

6 Conclusion

Starting from a deep analysis of the model watermarking scheme of [1], we find the fragility of the whole-image consistency is the root cause of why this watermarking framework cannot resist the data augmentation attack. To overcome this limitation, we propose a new watermarking methodology, "structure consistency", based on which a novel robust structure-aligned model watermarking algorithm is designed. Experiments demonstrate that the "structure consistency" can be utilized in both a global and local (semantic) way, and achieve much better robustness to data augmentation attacks and other adaptive attacks.

References

[1] J. Zhang, D. Chen, J. Liao, H. Fang, W. Zhang, W. Zhou, H. Cui, and N. Yu, "Model watermarking for image processing networks," in AAAI, vol. 34, no. 07, 2020, pp. 12 805–12 812.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in NeurIPS, 2012, pp. 1097–1105.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in NeurIPS, 2014, pp. 2672–2680.

[4] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in ICML. ACM, 2008, pp. 160–167.

[5] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in ICCV, 2015, pp. 2722–2730.

[6] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction apis," in USENIX Security, 2016, pp. 601–618.

[7] T. Orekondy, B. Schiele, and M. Fritz, "Knockoff nets: Stealing functionality of black-box models," in CVPR, 2019, pp. 4954–4963.

[8] Y. Uchida, Y. Nagai, S. Sakazawa, and S. Satoh, "Embedding watermarks into deep neural networks," in ICMR. ACM, 2017, pp. 269–277.

[9] Y. Adi, C. Baum, M. Cisse, B. Pinkas, and J. Keshet, "Turning your weakness into a strength: Watermarking deep neural networks by backdooring," in USENIX Security, 2018.

[10] J. Zhang, D. Chen, J. Liao, W. Zhang, H. Feng, G. Hua, and N. Yu, "Deep model intellectual property protection via deep watermarking," TPAMI, 2021.

[11] S. Szyller, B. G. Atli, S. Marchal, and N. Asokan, "Dawn: Dynamic adversarial watermarking of neural networks," arXiv:1906.00830, 2019.

[12] N. Lukas, Y. Zhang, and F. Kerschbaum, "Deep neural network fingerprinting by conferrable adversarial examples," arXiv preprint arXiv:1912.00888, 2019.

[13] H. Jia, C. A. Choquette-Choo, V. Chandrasekaran, and N. Papernot, "Entangled watermarks as a defense against model extraction," in USENIX Security, 2021, pp. 1937–1954.

[14] P. Maini, M. Yaghini, and N. Papernot, "Dataset inference: Ownership resolution in machine learning," arXiv preprint arXiv:2104.10706, 2021.

[15] Y. Li, L. Zhu, X. Jia, Y. Jiang, S.-T. Xia, and X. Cao, "Defending against model stealing via verifying embedded external features." AAAI, 2022.

[16] L. Charette, L. Chu, Y. Chen, J. Pei, L. Wang, and Y. Zhang, "Cosine model watermarking against ensemble distillation," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, no. 9, 2022, pp. 9512–9520.

[17] H. Wu, G. Liu, Y. Yao, and X. Zhang, "Watermarking neural networks with watermarked images," TCSVT, 2020.

[18] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," CVPR, 2017.

[19] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in CVPR, 2019, pp. 2337–2346.

[20] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in ICCV, 2017, pp. 2223–2232.

[21] J. Eusebio, H. Venkateswara, and S. Panchanathan, "Semi-supervised adversarial image-to-image translation," in International Conference on Smart Multimedia. Springer, 2018, pp. 334–344.

[22] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in MICCAI. Springer, 2015, pp. 234–241.

[23] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in ECCV. Springer, 2016, pp. 694–711.

[24] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in S&P. IEEE, 2019, pp. 707–723.