Can Image Watermarking Efficiently Protect Deep-Learning-Based Image Classifiers? – A Preliminary Security Analysis of an IP-Protecting Method

Jia-Hui Xie[1], Di Wu[2], Bo-Hao Zhang[3], Hai Su[4], and Huan Yang[5]

affiliation information masked as required by the anonymization guidelines

Abstract. Being widely adopted by an increasingly rich array of classification tasks in different industries, image classifiers based on deep neural networks (DNNs) have successfully helped boost business efficiency and reduce costs. To protect the intellectual property (IP) of DNN classifiers, a blind-watermarking-based technique that opens "backdoors" through image steganography has been proposed. However, it is yet to explore whether this approach can effectively protect DNN models under practical settings where malicious attacks may be launched against it. In this paper, we study the feasibility and effectiveness of this previously proposed blind-watermarking-based DNN classifier protection technique from the security perspective. We first show that, IP protection offered by the original algorithm, when trained with 256×256 images, can easily be evaded due to obvious visibility issue. Adapting the original approach by replacing its steganalyzer with watermark extraction algorithm and revising the overall training strategy, we are able to mitigate the visibility issue. Furthermore, we evaluate our improved approaches under three simple yet practical attacks, i.e., evasion attacks, spoofing attacks, and robustness attacks. Our evaluation results reveal that further security enhancements are indispensable for the practical applications of the examined blind-watermarking-based DNN image classifier protection scheme, providing a set of guidelines and precautions to facilitate improved protection of intellectual property of DNN classifiers.

Keywords: Blind watermarking · Intellectual property protection · Image steganography · Watermark extraction · Steganalysis · Evasion attacks · Spoofing attacks · Robustness attacks.

1 Introduction

As deep-learning-based image classification techniques continue to make exciting progress in miscellaneous application domains, ranging from medical image recognition [11, 12] to COVID-19 prevention [26], abuses of the copyrights of trained deep neural network (DNN) image classifiers have become a major concern hindering their widespread adoption [9, 19]. With the growing complexity of DNN image classifiers (e.g., in terms of model size and architectural complexity

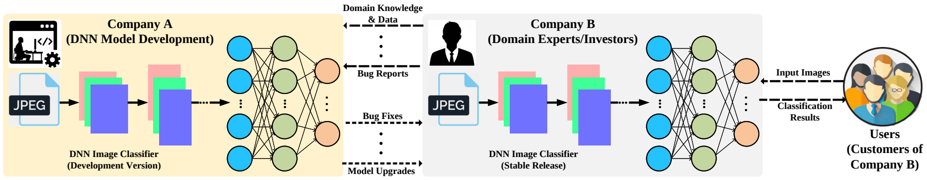


Fig. 1: A business case for DNN image classifiers where the right to use (also known as possession [15]) and ownership must be carefully distinguished.

[16]), time and computational resources dedicated to the training and fine-tuning of DNN models have been increasing rapidly. Consequently, copyright infringement targeting DNN image classifiers will result in an ever-growing loss to their legal owners, necessitating the development of techniques that protect the intellectual property (IP) of these classifiers.

In many practical applications of DNN image classifiers, drawing a distinction between the right to use (sometimes termed possession) and ownership is inevitable. Fig. 1 depicts a business case [15] where two companies A and B, one specialized in the design and implementation of DNN models while the other serving as a domain expert, work collaboratively to develop a DNN image classifier for profit. Company A may choose to lease its newly developed DNN image classifier to Company B under some contract clearly specifying the terms and scope of use. For instance, Company A may require that the model should only be used by Company B and should never be transferred to other companies without its explicit consent. Company B, while serving its customers for profit with the leased model, will report bugs to Company A, share newly acquired domain knowledge, and/or get feature/model upgrades from Company A. Oftentimes, Company A, as the developer (and ownership holder) of the DNN image classifier, may be unaware of the fact that his/her model, shipped and/or deployed without proper protection, can get stolen by adversaries through exploiting various mechanisms, such as electromagnetic side channel attacks [6] and model extraction attacks [10]. It is also possible that a certain malicious insider at Company B secretly shares the DNN image classifier with other companies, which violates the terms of use set by Company A. Without proper protection mechanisms in place, it is hard for Company A to detect/prove such violations, which will result in not only a loss to Company A but also a discouraging business atmosphere for other companies specialized in DNN model development.

To protect the copyrights of DNN image classifiers, an IP protection technique based on blind image watermarking is proposed in [19], which opens "backdoors" in the DNN image classifiers and enables model owners to externally verify their ownership. For instance, Company A in Fig. 2 can leverage this technique to train a DNN image classifier and then ship it to Company B. When a certain company other than Company B is suspected of illegally exploiting the DNN image classifier, Company A may externally verify whether an unmodified version of the model is being abused by sending image classification requests that can trigger the "backdoors" and verifying whether the embedded "backdoors" are indeed activated. However, this technique has not been thoroughly studied from

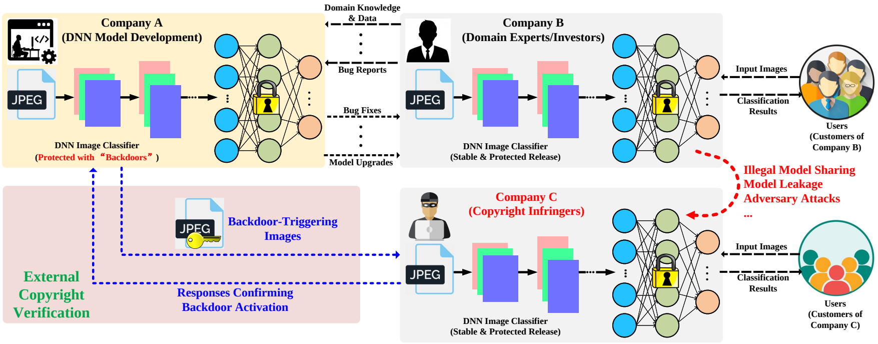


Fig. 2: External ownership/copyright verification enabled by image-watermarking-based techniques (e.g., ACSAC19 [19]) for DNN image classifiers.

the security perspective in practical context. The original version of this technique [19] will be called "ACSAC19" throughout this paper. In this paper, we re-examine the performance of ACSAC19, an end-to-end IP protection technique proposed in [19] and propose necessary enhancements to mitigate ACSAC19's visibility issue so that the external ownership verification operations can better evade the scrutiny of infringers. Furthermore, security analyses are conducted on our enhanced blind-watermarking-based IP protection techniques for DNN image classifiers, revealing the applicability and practicality issues of this DNN IP protection paradigm. The contributions of this paper can be summarized as follows:

- *Evaluation of a blind-watermarking-based IP protection technique (i.e., ACSAC19 [19]) for DNN image classifiers on a more practically-sized image dataset.* In contrast to the experiments conducted in [19] mainly on the CIFAR-10 dataset [18] consisting of tiny (32×32) images, we further evaluate ACSAC19 on the mini-ImageNet dataset [21, 30] consisting of more realistic 256×256 images. Our results show that the ACSAC19 approach [19] does not perform well in terms of invisibility on Mini-ImageNet, which may lead to evasion attacks that can easily be launched by infringers (e.g., through visual inspection).
- *Proposal of two enhanced versions of ACSAC19 that mitigate (in)visibility issue on mini-ImageNet.* Two enhanced versions of ACSAC19, namely end-to-end model with watermark extraction (termed "E2E-Extraction" approach throughout this paper) and two-phase host model fine-tuning (called "Two-Phase" approach throughout this paper), are proposed and evaluated. Our evaluation results show that the enhanced versions can mitigate the visibility issue, making it more practical for DNN image classifiers to adopt blind-image-watermarking-based IP protection techniques such as ACSAC19 [19].
- *A preliminary security analysis of blind-watermarking-based IP protection for DNN image classifiers based on our enhanced versions of ACSAC19.* Assuming that attackers may be able to gain access to miscellaneous sensitive information in the blind-watermarking-based IP protection process, we study whether our enhanced IP protection techniques, i.e., E2E-Extraction and

Two-Phase approaches, are vulnerable to evasion attacks, spoofing attacks, and robustness attacks. Our analyses identify critical information that must be kept secret from adversaries and generate useful guidelines on how the blind-watermarking-based IP protection paradigm, such as ACSAC19 and our enhanced versions, should be utilized in practical applications to protect the copyrights of DNN image classifiers.

2 Related Work

2.1 Protecting Deep Neural Network (DNN) Models from Copyright Infringements

As the business value of deep neural network (DNN) models continues to be substantiated by various successful applications ranging from medical image recognition [11, 12] to COVID-19 prevention [26], concerns on abuses of licensed DNN models have become a major issue, which not only undermines the business model of the artificial intelligence (AI) industry but discourages technological innovations as well [15]. To protect the copyrights of DNN models, intellectual property (IP) protection techniques have been proposed in recent years [35] for DNN models of different forms and objectives. For instance, the blind-watermarking-based IP protection method proposed in [19] is designed for DNN image classifiers. For DNN models that generate images as outputs (e.g., for tasks such as image segmentation), an IP protection framework utilizing watermarks encrypted by secret keys is proposed in [31]. To protect speech-to-text deep recurrent neural network models, a watermarking approach based on adversarial examples is devised in [24] to facilitate external verification of model ownership in a black-box manner. In [37], different types of watermarks are compared in the context of blind-watermarking-based DNN model protection. Recently, deep watermarking technique is applied to protect low-level image processing tasks (e.g., DNN backbones that automatically extract image features) against student-teacher learning [38]. In addition to IP protection through embedding watermarks into the data samples, other mechanisms, such as fingerprinting the classification boundaries of DNN models [8], embedding serial numbers to prevent unauthorized uses of models [29], deliberately rearranging DNN model's weights chaotically [20], and quantifying the similarities between victim and surrogate models [9], have been devised.

As IP protection methods for DNN models continue to improve and proliferate, security properties of these methods and possible attacks on them begin to draw the attention of both AI and security research communities. In [3], removal attacks on black-box backdoor watermarks protecting DNN model copyrights are reported, and successful watermark removal attacks on CIFAR-10 are demonstrated while maintaining sufficient host task performance (e.g., above 80% on CIFAR-10). Among recent studies on the security of blind-watermarking-based IP protection, ambiguity attacks, in which an adversary forges counterfeit watermarked images to undermine the reliability of the external ownership verification results, have shown to be a major issue. It is found in [13] that ambiguity attacks

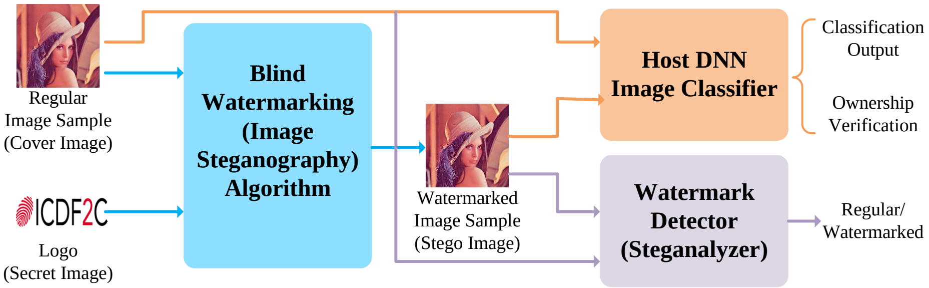


Fig. 3: The original end-to-end blind-watermarking-based IP protection method for DNN image classifiers proposed in [19]. Note that this technique is termed "ACSAC19" throughout this paper.

pose serious threats to existing DNN watermarking methods, and an enhanced watermarking strategy is proposed, which enables the protected models to reject counterfeit watermarked images. Meanwhile, the IP protection of generative adversarial networks (GANs) with possible presence of ambiguity attacks is also studied in [23].

Although blind-watermarking-based IP protection is recently studied extensively by the AI and security research communities, we observe that many of the existing results (e.g., [3, 13, 19, 23, 27, 37]) are primarily obtained using the CIFAR-10 dataset [18], which contains 32×32 images that seem to be overly small for both whole-image steganography and realistic image classification applications. It is hence necessary to further evaluate the performance of IP protection techniques for DNN models on datasets with more realistically sized images.

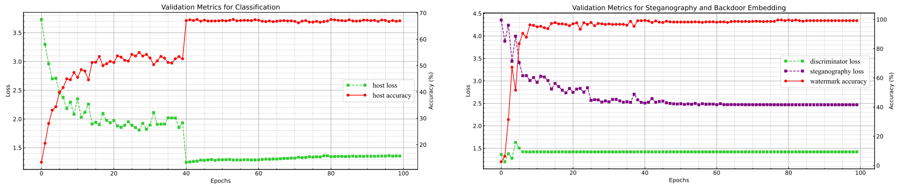
In this paper, we focus our attention on the IP protection of DNN image classifiers using the ACSAC19 method [19] in practical contexts from the security perspective. Fig. 3 presents the ACSAC19 method originally proposed in [19]. In the ACSAC19 method, whole-image steganography, which embeds secret images into cover images as blind watermarks, is leveraged to generate stego images (also known as "trigger" images) that will be presented to the host DNN image classifier to verify ownership. During the end-to-end training process of the ACSAC19 approach, the steganalyzer detects whether images presented to the host model contain blind watermarks or not, interacting with the steganography module to ensure invisibility of the watermarks. The host DNN is the target model to be protected, and ACSAC19's end-to-end method trains the host DNN in such a way that it will generate correct classification outputs for regular images and produce ownership verification outputs for the stego images. In ACSAC19, the ownership verification outputs are random numbers associated with individual stego images. Note that both the host DNN image classifier and the steganalyzer interact with the blind watermarking module throughout the end-to-end training process, which strives to generate stego images that can evade the scrutiny of the steganalyzer while being correctly recognized by the host DNN image classifier as "trigger" images. Although satisfactory IP protection performance is achieved on CIFAR-10, we note that the ACSAC19 approach has not been thoroughly examined in practical settings from the security perspective.

2.2 Whole-Image Steganography and Steganalysis

Whole-image steganography [28] refers to the process of hiding a secret image into a cover image so that it is visually hard to tell whether an arbitrary image is a stego image (i.e., image containing a blind watermark) or not. In contrast to conventional message steganography algorithms based on the least-significant-bit (LSB) approach [22], whole-image steganography naturally requires a larger information hiding capacity, and deep-learning-based approaches typically outperform conventional methods. In [4, 5], a set of convolutional neural networks (CNNs) are combined into an end-to-end model to achieve large capacity information hiding required by whole-image steganography. In [25], a CNN-based encoder-decoder network is designed to embed secret images as blind watermarks into cover images. Recently, HiNet [17] leverages invertible neural network (INN) architecture, discrete wavelet transform (DWT), and inverse DWT (IDWT) to construct a whole-image steganography model that outperforms peer models in terms of secret image recovery. We note that deep-learning-based whole-image steganography algorithms typically offer a pair of models, one for hiding the secret images and the other for extracting them.

In contrast to blind-watermarking-based IP protection for DNN models, deep-learning-based steganography algorithms (e.g., [4, 17, 25]) are all evaluated on realistic image datasets such as ImageNet. To facilitate practical applications of existing IP protection techniques in production systems based on DNN image classifiers, it is necessary to re-examine their feasibility and performance on realistic datasets with images of reasonable sizes. Furthermore, it has been well understood that it is generally hard for image steganography to balance among invisibility, security, information hiding capacity, and robustness [4, 28]. Therefore, it is also important to further evaluate existing blind-watermarking-based IP protection techniques from the perspective of steganography algorithm performance.

The security of a steganography algorithm refers to how hard it is to detect that a certain secret image is embedded, and the detector is known as a steganalysis algorithm (or a steganalyzer). Proposed in [34] and evaluated in [33], XuNet consists of a group of CNNs and can effectively tell whether some secret is hidden in a given image. In [36], the YeNet steganalyzer is proposed to directly learn hierarchical representations of images using CNNs. In addition, to thoroughly examine "noise residuals" where secret information may be embedded, SRNet [7] is proposed to construct deep residual network and detect previously suppressed stego signals. In [32], a CNN-based steganalyzer for content-adaptive image steganography in the spatial domain is proposed. We note that the various steganalyzers developed by the security research community may be exploited by adversaries to deter model owners from externally verifying their ownership. Hence, it is important to investigate whether such tools can be exploited and to what extent existing blind-watermarking-based IP protection approaches based on image steganography can be jeopardized.



(a) Host DNN image classifier (ResNet-18) performance.

(b) Performance of steganography and steganalysis algorithms.

Fig. 4: End-to-end training of the originial ACSAC19 method [19]. Note that all the performance results are obtained on the validation set throughout training and that the host model is ResNet-18.

3 Enhancing a Blind-Watermarking-Based IP Protection Technique (ACSAC19) for DNN Image Classifiers

3.1 Re-examining the Original End-to-End Blind-Watermarking Method (ACSAC19) for DNN Image Classifier Protection

In this section, we re-examine the steganography performance of the ACSAC19 approach [19] on the mini-ImageNet dataset [21, 30]. We use the authors' implementation [39]. The host model to be protected is ResNet-18 [14], which is also used in the evaluation of ACSAC19 in [19].

Experiment Settings. To evaluate ACSAC on a more practically sized image dataset, we construct the mini-ImageNet dataset without downsizing the images using an open-source tool [21]. The resultant min-ImageNet dataset includes 60,000 256×256 images, evenly drawn from 100 classes (i.e., 600 images per class in our 256×256 image dataset). We perform a $8:1:1$ split on the dataset, with 80% of that images dedicated to training, 10% for validation, and the remaining 10% for testing. We note that the same dataset split will be applied to other experiments throughout this paper.

We closely follow the ACSAC19 training process outlined in [19] and implemented in [39]. Fig. 4 depicts the training process and shows that all three modules of ACSAC19, namely the host model, the steganography algorithm, and the steganalyzer, are trained until convergence. At the end of the training process, the accuracy of the ownership verification task for the host model settles above 92% on the testing set, while the performance of ResNet-18 (i.e., the image classification task) does not obviously deteriorate. Although this result is consistent with those reported in [19], we observe that ACSAC19 does not visually perform well on mini-ImageNet: Fig. 5 includes a set of cover images and the corresponding stego images generated by ACSAC19 trained on mini-ImageNet. Despite the deployment of a steganalyzer to enhance steganography invisibility, the steganography module of ACSAC19 trained on mini-ImageNet does not perform well in terms of (in)visibility. All stego images in Fig. 5 contain obvious visual defects. Take the leftmost image pair in Fig. 5 as an example. The facial part of the male singer in the cover image does not include obvious

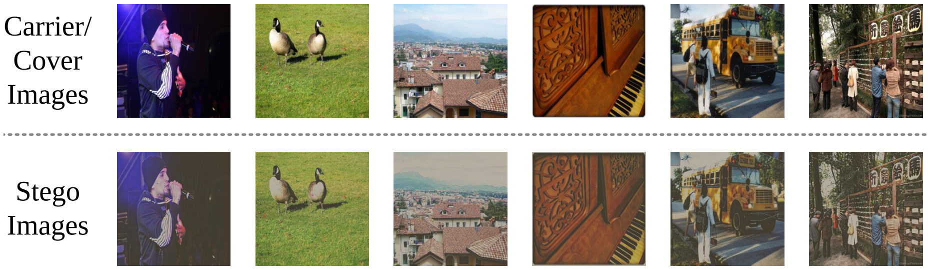


Fig. 5: Visual defects generated by the ACSAC19 approach [19] on the mini-ImageNet dataset. Note that secret images (i.e., watermarks) are randomly selected from mini-ImageNet.

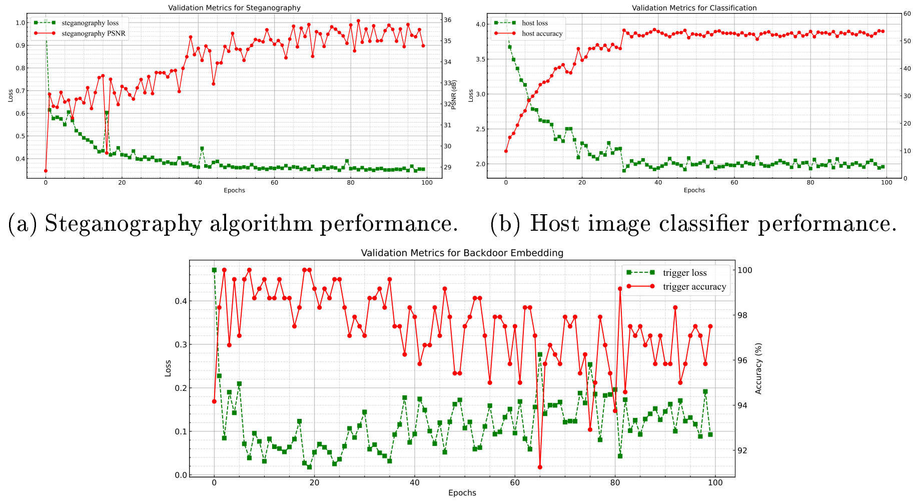
defects, whereas the corresponding part in the stego image contains evident defects that may alert an image inspector (e.g., the copyright infringer). Such a visibility issue significantly undermines ACSAC19's main objective of blind-watermarking-based IP protection, i.e., external ownership verification: Let us consider the ACSAC19 approach illustrated in Fig. 3. If the stego images generated by the blind-watermarking algorithm, which have been properly learned by the host DNN during end-to-end training, can easily be visually discerned from regular image samples, external ownership verification relying on the outputs of the protected host DNN becomes infeasible. When the model owner presents the stego images generated by ACSAC19, the copyright infringers can easily pick them out through visual inspection, leading to the failure of external ownership verification. Therefore, for the ACSAC19 approach to be useful in realistic image classification applications with images larger than those in CIFAR-10, our results suggest that further enhancements to it are required to at least mitigate this issue of watermark detectability.

3.2 Enhancing a Blind-Watermarking-Based IP Protection Technique (ACSAC19) for DNN Image Classifiers

To address the detectability issue of the ACSAC19 method [19], we propose two alternative approaches, i.e., an enhanced end-to-end with watermark extraction (called the "E2E-Extraction" method) and a two-phase host fine-tuning approach (termed the "Two-Phase" method).

End-to-End Blind-Watermarking IP Protection with Secret Image Extraction. One of our enhancements still leverages the end-to-end training method presented in [19] and illustrated in Fig. 3 but replaces the pair of steganography algorithm and steganalyzer with the a pair of steganography algorithm and the corresponding secret image extraction algorithm. We call this enhanced version of ACSAC19 the E2E-Extraction approach. Note that the end-to-end training strategy as well as loss function in E2E-Extraction remain the same as ACSAC19 [39].

The rationale behind the E2E-Extraction approach is that the steganalysis in ACSAC19, which gives a binary output about whether some secret is hidden



(a) Steganography algorithm performance. (b) Host image classifier performance.

(c) Ownership verification performance of host classifier.

Fig. 6: End-to-end training of the proposed E2E-Extraction IP protection approach. Note that all the performance results are obtained on the validation set throughout training and that the host model is ResNet-18.

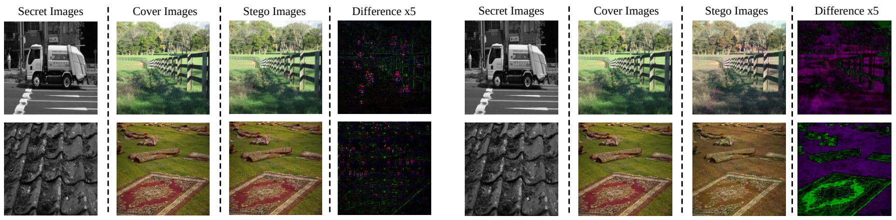
in a given image, may converge much earlier than the steganography algorithm, leading to (in)visibility/detectability issue. The steganography algorithm must strive to achieve two goals: On the one hand, it must not hide the secret images too well. Otherwise, the host model may not be able to associate the stego images with ownership verification labels. On the other hand, it should hide the secret images well enough so that the steganalyzer is not able to recognize them with high probability. ACSAC19 was able to achieve such a balance on CIFAR-10, but it fails to do so on more practically sized datasets, such as mini-ImageNet. This is substantiated by Fig. 4b, which shows that the steganalyzer (i.e., the discriminator) is able to converge after the first few epochs. During these epochs, Fig. 4b shows that all three modules of ACSAC19, i.e., the steganography algorithm, the host DNN image classifier, and the steganalyzer, actively interact with each other. However, as the steganalyzer converges, the loss function design of ACSAC19 does not demand it to further interact with the other two modules. As shown in both Fig. 4a and Fig. 4b, the steganography algorithm and the host model further interact toward convergence till around the 40^{th} epoch, and the steganalyzer does not effectively participate in this process after the first few epochs. As a result, the steganalyzer hardly provides further useful feedback to the steganography algorithm, which leads to visibility issue (see Fig. 5).

Unlike the steganalyzer of ACSAC19 which is designed to function as a pair with the steganography algorithm in ACSAC19, existing watermark (or secret image) extraction algorithms, on the other hand, have been trained together with the corresponding steganography algorithms as pairs in prior work (e.g., [4, 5, 25, 17]. Since the ACSAC19 approach employs the steganalyzer with the steganography algorithm to form an generative adversarial subnetwork [19], our

E2E-Extraction approach essentially pushes the adversarial training idea further by letting a more specialized model capable of extracting the embedded secrets interact with the steganography module.

To evaluate the performance of our proposed E2E-Extraction technique, we choose the pair of steganography and watermark extraction algorithms proposed in [25] and follow the end-to-end training procedures outlined in [38]. The settings of our experiment are the same as those described in Sec. 3.1. Fig. 6 summarizes the performance of the three components (i.e., the steganography module, the host DNN image classifier, and the stego image extractor) in our E2E-Extraction technique on the validation set throughout the training process. As shown in Fig. 6c, stego images (i.e., "backdoor" images embedded into the host model through end-to-end training) can effectively "trigger" ownership verification outputs at the host model, achieving satisfactory verification accuracy (i.e., "trigger accuracy" in Fig. 6c). It can be observed that all three components are able to converge and achieve satisfactory performance at the end of the training process and that the ownership verification accuracy stays above 95% most of the time. Compared to the ACSAC19 approach, both the steganography algorithm (see Fig. 6a) and the host model (see Fig. 6b and Fig. 6c) interact with the other modules more actively (and thus more fluctuations), leading to improved visual quality of the stego images.

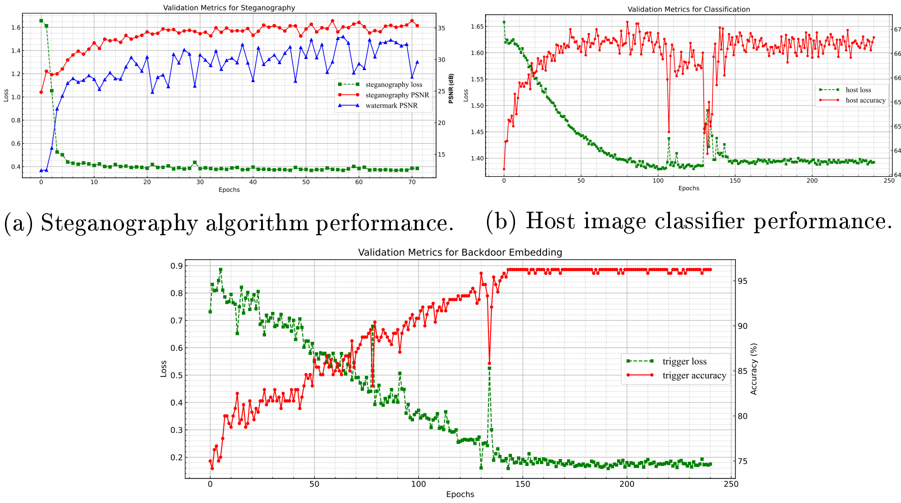
Fig. 7a shows two group of images used and generated by the steganography algorithm we use in E2E-Extraction training. It can be observed that the stego images indeed have better visual quality. In contrast to the ACSAC19 approach [19, 39] trained on mini-ImageNet, our proposed enhancement effectively mitigates the issue of watermark detectability and are hence better-suited for protecting copyrights of DNN image classifiers. However, it should be noted that visual defects still exist and that a careful inspector (or a copyright infringer exploiting the widely used "different×5" approach) might still be able to identify images sent by the model owner for the purpose of external ownership verification.



(a) Sample images used and generated by E2E-Extraction.

(b) Sample images used and generated by Two-Phase approach.

Fig. 7: Sample secret, cover, and stego images of the E2E-Extraction and Two-Phase approaches. Note that the "Difference x5" image, which is often used to help quickly detect blind watermarks, is generated by taking the difference between the cover and the stego images and then multiplying the results by 5.



(a) Steganography algorithm performance. (b) Host image classifier performance.

(c) Ownership verification performance of host classifier.

Fig. 8: Performance of the proposed Two-Phase IP protection approach. Note that all the performance results are obtained on the validation set throughout the respective training phases.

Two-Phase Approach with Host Image Classifier Fine-Tuning. The other enhanced IP protection technique we propose is constructed as follows: First, the host DNN image classifier is trained separately to ensure that it offers satisfactory performance on its main task (i.e., image classification). Meanwhile, we choose a pair of steganography algorithm and secret image extraction algorithm, which can be regarded as our proposed alternatives to the steganography and steganalysis algorithms in ACSAC19 (see Fig. 3). Next, a set of images are randomly selected from the mini-ImageNet dataset to form the cover image set and secret image set. The steganography algorithm and the corresponding watermark extraction algorithm are then trained together using the cover and secret image sets. Using the trained steganography algorithm, we generate a third image set, which is the stego image set. Finally, we mix the stego image set with the training set of the host model and fine-tune the host model in such a way that, in addition to generating correct classification outputs for regular images, stego images will be recognized and proper ownership verification outputs will be generated. This approach is call Two-Phase approach because it virtually involves two phases, namely the preparation phase and the host fine-tuning phase. In the preparation phase, the host model as well as the pair of steganography and watermark extraction algorithms are trained separately. In the host fine-tuning phase, we further train the host model with the stego images (i.e., host model "backdoor" images) to enable external ownership verification while maintaining its performance on the main task.

To evaluate the performance of our proposed Two-Phase approach, we use the same experiment settings as described in Sec. 3.1. The steganography and watermark extraction algorithm pair remains to be the one proposed in [25] and the host classifier is still ResNet-18. Fig. 8 summarizes the performance results of the major components of our Two-Phase method: As shown in Fig. 8a, the trained steganography and stego image extraction algorithms perform satisfactorily well. Since the watermark extractor achieves stego image extraction performance (i.e., “watermark PSNR” in Fig. 6a) of more than 26 dB (the extracted watermark images have acceptably good visual quality as reported in [25] at this PSNR level), it can be leveraged to further prove ownership of the host model: After external ownership verification is successfully conducted (i.e., the queried host model generates the expected ownership verification outputs), the model owner can further extract the blind watermarks to prove ownership, possibly in front of a jury or notary. We also note that the host model performance on the main task (i.e., image classification) reported throughout this paper (e.g., see Fig. 6b and Fig. 8b) is reasonable according to [1, 2]. Note that the Two-Phase approach trains the steganography algorithm and the watermark extraction algorithm as a pair in the preparation phase, whereas the host model is independently trained during this phase. As shown in Fig. 8a, the peak signal-to-noise ratios (PSNR) for both the steganography algorithm and the watermark extractor are sufficiently high after about 70 epochs. However, since the host model does not interact with the pair of steganography algorithm and watermark extractor during the preparation phase, it host model fine-tuning task requires more epochs because the steganography algorithm and the watermark extractor are able to collaboratively train each other to an extent that can be challenging for the host model. In fact, the Two-Phase approach requires careful selection of hyperparameters such as learning rate, which leads to a longer model development cycle.

Fig. 7b shows the images used and generated by the steganography algorithm we use in the Two-Phase training process. It can be observed that, though the colors of the stego images are slightly distorted, both stego image samples have better visual quality. In contrast to the ACSAC19 approach [19, 39], our two-phase approach can also alleviate the issue of watermark detectability, making it more suitable for the protection of DNN classifier copyrights. It should be noted that, in practical applications of IP protection of DNN image classifiers, the original cover images may not be presented to the host model at all, so the copyright infringers will not be able to tell whether the stego images contain obvious color distortion. In fact, if the copyright infringers do not have access

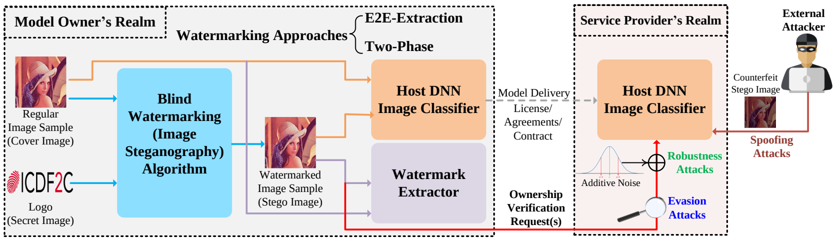


Fig. 9: The overall scheme for a preliminary security analysis of our enhanced blind-watermarking-based IP protection techniques of DNN image classifiers.

to the cover images, it will also be hard for them to conduct the "difference×5" operation.

4 A Preliminary Security Analysis of Our Enhanced Blind-Watermarking-Based IP Protection Techniques for DNN Image Classifiers

To facilitate applications of blind-watermarking-based IP protection in practical image classification systems, further analysis of its security performance on realistic image datasets (e.g., mini-ImageNet) is of urgent necessity. We consider the application scenarios illustrated in Fig. 9: The model owner leverages some blind-watermarking-based IP protection technique (e.g., E2E-Extraction or Two-Phase approach introduced in Sec. 3) to protect his/her newly developed DNN image classifier. The classifier is then delivered to a service provider, who agrees to utilize it within the constraints specified by a certain license or contract. However, it is possible that a certain insider may purposely leak the model, and the service provider himself/herself may also violate the license/contract terms and abuse the model (e.g., providing services to a competitor of the model owner). When the model owner becomes aware of a potential violation of the license/contract terms, he/she can pretend to be a customer requesting (illegal) model service from the service provider. Leveraging the stego images generated during the training process that opens "backdoors" to the host model through blind watermarking, the model owner can externally verify model ownership, allowing him/her to take further actions and protect his/her copyright.

Attacks on Blind-Watermarking-Based IP Protection. However, to ensure the effectiveness and reliability of the blind-watermarking-based IP protection paradigm, we note that its security properties under various attacks, should be carefully examined. As depicted in Fig. 9, our preliminary analysis proposes that the following three types of attacks should be considered when the security of blind-watermarking-based IP protection of DNN image classifiers is examined:

- *Evasion attacks.* This attack may be launched by the service provider who abuses the protected host model. Suppose that visual inspection and/or a steganalyzer may be employed by the service provider to examine every image samples passed to his/her copy of the host model. If a stego image is identified, the service provider can simply reject the corresponding request and do not pass the detected stego image to the host model. In this way, external ownership verification must be repeated for multiple times and may eventually fail if the attacker's steganalyzer is strong enough.
- *Spoofing attacks.* This attack is similar to the ambiguity attack described in Sec. 2.1 and is realized by an external attacker sending counterfeit stego images to the user's copy of the host model for ownership verification. The counterfeit stego images are produced by the attacker. Evidently, the more

is learned about the steganography algorithm employed by the model developer, the better the counterfeit stego images can be. This attack may be launched to serve one of the following purposes: On the one hand, if the attack succeeds in falsely claiming ownership of the user's copy of the host model, he/she may gain illegal financial benefits. On the other hand, even if the attack is detected, possibly via other measures such as verifying the identity of the attacker, the reliability of the external ownership verification enabled by blind-watermarking-based IP protection is undermined.
- *Robustness attacks.* Steganography algorithms are susceptible to robustness attacks [28]. For instance, the watermark extractor may fail if the stego images are severely distorted during transmission. Consequently, IP protection based on blind image watermarking should also be evaluated against robustness attacks: The service provider who abuses the host model may choose to add noise or distort all the images sent to his/her copy of the host model, in hope that such image manipulations will cause the steganography system to fail while exploiting the typically better robustness of the main task of the host model.

Assumptions on Attacker's Capabilities. In addition to the types of possible attacks that may be launched in practical settings, our preliminary security analysis also takes the attacker's capabilities into account:

- *Naive external attackers.* A naive attacker external to the model owner's realm (see Fig. 9) does not have access to much detail about how the blind-watermarking-based IP protection mechanism is constructed and trained. We assume that such an attacker may be able to learn about the fact that a certain blind watermarking technique is adopted to protect the intellectual property of the host DNN image classifier he/she obtains. However, such an attacker does not know the exact steganography algorithm employed by the model owner.
- *Sophisticated external attackers.* A sophisticated attacker external to the model owner's realm (see Fig. 9) may conduct reconnaissance and eventually learn about certain information on the design and training strategy of the blind-watermarking-based IP protection method. For instance, it is likely that such an attacker is able to find out the name of the steganography algorithm utilized by the model owner in IP-protecting training of the host model. It is also likely that such an attacker is able to get hold of a subset of the secret images used by the model owner (e.g., by analyzing external ownership verification requests previously issues by the model owner). However, it is generally hard for such an external attacker to gain access to the weights of the trained steganography algorithm.
- *Malicious insiders.* An insider, or a sophisticated attacker assisted by an malicious insider within the model owner's realm (see Fig. 9), may be able to obtain the trained version of the steganography algorithm and the watermark extractor. For steganography algorithms, it is also possible to obtain their weights by obtaining the dataset used in the training process [28].

In the remainder of this paper, we will conduct our preliminary security analysis and examine the security performance of E2E-Extraction and Two-Phase approaches, which will both help AI practitioners better assess the applicability of blind-watermarking-based IP protection and reveal how existing IP protection techniques may be further improved.

5 Launching Evasion Attacks on Blind-Watermarking-Based Image Classifier Protection Techniques

Assuming that a copyright infringer will at least visually inspect the images submitted to his/her copy of the host model, We evaluate the security performance of our proposed E2E-Extraction and Two-Phase techniques, which have been shown to exhibit satisfactory invisibility performance. We choose two steganography algorithms and their corresponding stego image extraction algorithms proposed in [4, 25] to implement our E2E-Extraction and Two-Phase IP protection mechanisms. In the remainder of this paper, the steganography algorithm in [25] is called the "Encoder-Decoder" (En2D) model in our experiments, while the steganography algorithm in [4] is called "GglNet".

Observing the fact that our Two-Phase approach generates stego images with better invisibility, we first study its security performance under evasion attacks. Our experiments are conducted as follows: First, we set the learning rate for the En2D model to 10^{-3} and the batch size to 30. Four random seeds, i.e., $2022, 1211, 204$ and 109, are chosen to train four different versions of the En2D steganography algorithm until convergence, and we name them En2D-2022, En2D-1211, En2D-204, and En2D-109, respectively. As for GglNet, we set its learning rate to 10^{-4} and the batch size to 20. A random seed of 2022 is chosen. For both steganography algorithms, we use the widely-used mean-squared-error (MSE) loss function and the Adam optimizer. The stego image extractors are trained in pairs with the corresponding steganography algorithms. The host

Table 1: Security performance of the proposed Two-Phase approach under evasion attacks launched by naive and sophisticated external attackers. The higher the detection/evasion rate, the easier it is for an attacker to evade external ownership verification.

Attacker Type	Steganalyzer	Attacker Steganography Algorithm	Model Owner Steganography Algorithm	Detection/Evasion Rate			
				Without Dataset Overlaps		With Dataset Overlaps	
				Designed	Actual	Designed	Actual
Naïve External Attackers	SRNet	GglNet	EnD-2022	86.30%	58.83%	85.55%	56.12%
		EnD-2022	GglNet	81.00%	47.08%	82.45%	49.29%
	YeNet	GglNet	En2D-2022	77.65%	34.42%	76.5%	40.38%
		En2D-2022	GglNet	91.15%	28.12%	87.60%	44.33%
Sophisticated External Attackers	SRNet	En2D-1211	En2D-2022	83.00%	49.54%	84.70%	48.83%
		En2D-204	En2D-2022	99.70%	47.88%	86.45%	49.67%
		EnD-109	En2D-2022	67.85%	50.83%	67.05%	52.88%
	YeNet	En2D-1211	En2D-2022	91.75%	46.29%	92.05%	53.46%
		En2D-204	En2D-2022	91.30%	44.96%	90.50%	51.25%
		EnD-109	En2D-2022	72.90%	67.92%	75.85%	76.67%

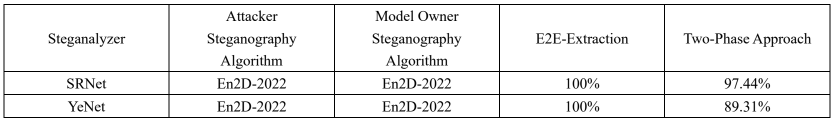
model (i.e., ResNet-18) is also optimized for its main task at this phase. Next, we complete the host model fine-tuning step for ownership verification using the stego images generated by different steganography algorithms. To launch evasion attacks, we choose two deep-learning-based steganalyzers, i.e., SRNet [7] and YeNet [36]. To simulate different assumptions on attacker's capabilities, we separately train new steganography models and leverage them to generate datasets for the steganalyzers, which are trained in three different manners:

- *Independently training a steganography algorithm with or without coincidental dataset overlaps.* In this case, we assume that evasion attacks are launched by a naive external attacker, who does not know the exact steganography algorithm used in the IP-protecting training process. In our experiments, such attacks are simulated by evasion attacks implemented with steganography algorithms different from the one used by the model owner. We note, however, since our images are drawn from the publicly available mini-ImageNet dataset, we also consider the scenarios where a small portion ($<5\%$) of the images randomly selected by the attacker coincides with the secret and cover images selected by the model owner.
- *Training the same steganography algorithm without sharing any model weights and/or parameters.* In this case, we assume that evasion attacks are launched by sophisticated external attackers. In our experiments, such attacks are simulated by evasion attacks implemented using the same steganography algorithm as the model owner, but with a different random seed (and hence different model weights at convergence).
- *Training the same steganography algorithms with the same data sets.* In this case, evasion attacks are launched by a malicious insider. In our experiments, we train the steganalyzers using exactly the same datasets (i.e., cover, secret, and stego image sets) as the model owner.

Table 1 summarizes the security performance of our Two-Phase technique under evasion attacks launched by external attackers. Note that the "designed" column reports the steganalyzers' performance of detecting stego images on the testing sets prepared by the attackers. However, since external attackers are not able to learn about internal details of the steganography algorithms used by the model owner, the actual detection is significantly lower. Table 1 suggests that an external attacker can launch evasion attacks to raise the barrier to external ownership verification: Even for the naive attacker launching evasion attacks with En2D-2022 and YeNet, the model owner must, with a non-negligible probability, prepare more watermarked images to successfully claim copyrights. However, the number of stego images for external ownership verification is fixed once the Two-Phase approach completes training. Even though it is still possible to utilize the steganography algorithm in the Two-Phase approach to generate new stego images, we note that these newly generated images have not been presented to the host image classifier for fine-tuning and the chance of failure to verify ownership is higher with these stego images.

Since coincidental overlaps are rare in practical applications (where the model owner may not disclose his/her private secret and cover image sets at all), we

Table 2: Security performance (detection/evasion rates) of the proposed E2E-Extraction and Two-Phase methods under evasion attacks launched by malicious insiders. The higher the detection/evasion rate, the easier it is for an attacker to evade external ownership verification.



Steganalyzer	Attacker Steganography Algorithm	Model Owner Steganography Algorithm	E2E-Extraction	Two-Phase Approach
SRNet	En2D-2022	En2D-2022	100%	97.44%
YeNet	En2D-2022	En2D-2022	100%	89.31%

can also observe that coincidental overlaps of secret and cover image sets do not help much in boosting the detection rate in an evasion attack. This also suggests that knowledge about the internals of the steganography algorithm employed by the model owner can only be derived from the stego image set (an observation in consistence with the results and discussion in [28, 34, 33, 36]), which should be kept secret by the model owner. It should also be noted that the actual detection rate achieved by the sophisticated external attackers is above 44%. If such an attacker colludes with other compromised service providers using the same host model to collect stego images sent by the model owner, evasion attacks will help these adversaries exhaust the stego image set more quickly. Security performance of the E2E-Extraction approach under evasion attacks launched by external attackers exhibits characteristics similar to those of the Two-Phase method. We will report detailed results in a separate technical report due to space constraints.

We then examine the security performance of both enhanced methods under evasion attacks launched by a malicious insider. Table 2 summarizes our evaluation results and shows that it is possible for a malicious insider to evade most ownership verification attempts of the model owner. Therefore, in addition to securing the one-to-one correspondence between secret image and model owner [13, 35], it is also necessary to ensure that success rate of evasion attacks should be kept reasonably low. Special care must be taken to protect steganography algorithms (especially the model weights) trained with the E2E-Extraction or Two-Phase approach from insider attackers, which can help prevent significant loss to the model owner and keep the IP protection mechanism reliable.

6 Launching Spoofing Attacks on Blind-Watermarking-Based Image Classifier Protection Techniques

As revealed in Sec. 5 and in [13], the host DNN image classifier in our E2E-Extraction and Two-Phase approaches are trained to recognize stego images prepared and presented by the model owner. In the experiments in Sec. 5, we have also trained the host model on the ownership verification task while maintaining their performance on the main task. To evaluate the security performance of our proposed enhancements to ACSAC19 under spoofing attacks, we train multiple steganography algorithms independently from the blind-watermarking-based IP protection training process.

Table 3: Security performance (success rates of external ownership verification) of the proposed E2E-Extraction and Two-Phase methods under spoofing attacks (Orange: attacks launched by an insider; Blue: attacks launched by naive external attackers; White: attacks launched by sophisticated external attackers.).

IP Protection Framework	Owner’s Attacker’s	En2D-109	En2D-204	En2D-1211	En2D-2022	GglNet
Two-Phase	En2D-109	99.6667%	1.3333%	99.6667%	99.7500%	2.5417%
	En2D-204	3.0833%	99.2083%	3.2917%	3.7083%	47.7500%
	En2D-1211	98.6250%	4.6667%	99.2917%	99.4167%	0.5833%
	En2D-2022	98.1667%	5.6667%	98.7500%	99.3333%	0.5000%
	GglNet	2.3333%	59.6250%	2.2917%	2.2917%	88.5417%
E2E-Extraction	En2D-2022 (JPG)	0.4167%	7.4167%	0.4167%	0.4583%	2.3750%
	Eb2D-2022 (MAT)	0.6667%	14.9583%	0.8333%	0.5417%	7.2083%

Table 3 presents the rates at which counterfeit stego images generated by an attacker are confused by the host DNN model with genuine stego images generated by the model owner. The values with a blue background are obtained under attacks initiated by a naive external attacker. Although the spoofing attack may opportunistically succeed, it is in general hard for a naive attacker to train a steganography algorithm with output stego images that are easily confused with those prepared by the model owner.

The values marked with an orange background are obtained under the assumption that the attacks are launched by a malicious insider. We note that, although such attacks can easily succeed, the resources required to collect a sufficiently large portion (e.g., >90%) of the model owner’s stego image set can be prohibitively expensive. The values in Table 3 with a white background are obtained assuming that spoofing attacks are launched by a sophisticated external attacker. Obviously, it is possible that a malicious insider or a sophisticated external attacker can compromise the Two-Phase approach (e.g., the external attacker can train multiple steganography algorithms and see whether one of them can give a relatively high success rate of ownership verification). Hence, even a sophisticated external attacker, who is outside the model owner’s realm, may be able to falsely claim model ownership and/or undermine the credibility of blind-watermarking-based IP protection.

Moreover, we note that an attacker can invest further on a particular steganography algorithm to opportunistically boost the success rate of spoofing attacks. Take the En2D models as an example. If the sophisticated attacker is able to confirm that some version of the En2D model is used by the model owner during IP-protecting training, then he/she can train a series of En2D models and try to find out the combination of model versions offering high success rate for spoofing attacks. Such knowledge can be reused to attack multiple host models protected by the Two-Phase or E2E-Extraction approach. As shown in Table 3, the attacker do not need to find the exact version of the En2D model (or obtain the stego image set used by the model owner). If En2D-109 is employed by the model owner, the attacker implementing En2D-1211 or En2D-2022 will be able to successfully launch spoofing attacks. Therefore, in practical applications of our enhanced versions of ACSAC19, it is necessary for the model owner to conduct similar experiments in advance and verify that the steganography algo-

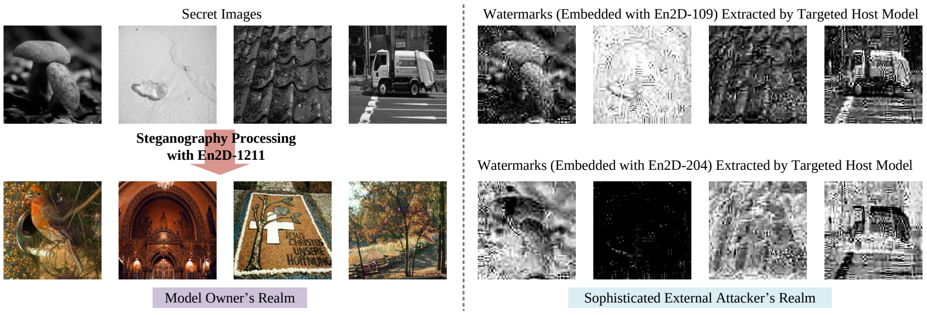

Secret Images
Steganography Processing
with En2D-1211
Model Owner's Realm
Watermarks (Embedded with En2D-109) Extracted by Targeted Host Model
Watermarks (Embedded with En2D-204) Extracted by Targeted Host Model
Sophisticated External Attacker's Realm

Fig. 10: Spoofing attacks launched by an external attacker who is able to obtain a subset of the model owner's secret images. Note that En2D-1211 is employed by the model owner, and the attacker attempts to launch spoofing attacks with En2D-109 and En2D204.

Table 4: Performance (success rates) of the ownership verification task of the proposed E2E-Extraction and Two-Phase approaches under robustness attacks.

IP Protection Framework	Steganography-Host Pair	Gaussian Noise	Salt & Pepper Noise	Noiseless
Two-Phase Approach	En2D-109-ResNet18	84.8750%	96.7083%	99.6667%
	En2D-204-ResNet18	87.9167%	98.6250%	99.2083%
	En2D-1211-ResNet18	85.4583%	98.5417%	99.2917%
	En2D-2022-ResNet18	93.2500%	98.9167%	99.3333%
	GglNet-ResNet18	70.9167%	85.2917%	88.5417%
E2E-Extraction	En2D-2022-ResNet18	80.0417%	83.2917%	87.6667%

rithm he/she chooses offers a sufficiently high degree of security under possible spoofing attacks.

Finally, Fig. 10 depicts how an external attacker with access to a subset of the model owner's secret images may be able to work out a plan to falsely claim model ownership. As shown in this figure, the attacker may not get a high success rate if he/she starts with En2D-204. However, if the attacker is patient enough to try multiple random seeds, it is possible that he eventually finds En2D-109, which is good enough for the purpose of spoofing attacks. Due to space constraints, further visual results will be reported in a separate technical report.

7 Launching Robustness Attacks on Blind-Watermarking-Based Image Classifier Protection Techniques

In practical applications of DNN image classifiers, additive noise may be introduced during transmission. Evidently, additive noise will impact not only the ownership verification task but the main task of the host model as well. In this paper, we launch robustness attacks on our proposed E2E-Extraction and Two-Phase techniques while ensuring the that performance of the main task does not obviously deteriorate. We choose to examine such a configuration because in practical settings, the attacker (i.e., the copyright infringer abusing the protected host DNN image classifier) would like to launch robustness attacks both to invalidate external ownership verification tests issued by the model owner and to utilize the host DNN classifier for profit.

In our experiments examining the impacts of robustness attacks, we assume that robustness attacks are launched by a naive external attacker. Although malicious insiders or sophisticated external attackers may also exploit robustness attacks, our assumption is reasonable because robustness attacks present a relatively low technical barrier: The attacker does not need to be an expert on whole-image steganography, so it is more practical for a naive external attackers to first consider such attacks.

Table 5 shows that both Gaussian noise (with a mean of 30, a standard deviation of 15 for 8-bit pixels) and salt & pepper noise (with $1,200$ points per image) will slightly deteriorate the performance of the main task. Meanwhile, as shown in Table 4, Gaussian noise is able to further deteriorate host model performance on the ownership verification tasks.

Based on these observations, we argue that for more sophisticated host models (e.g., ResNet-101), more noise can be added to all the input images to the host model by the external attackers (because the host model will have more representative power than the steganography module, and thus more likely to remain robust). To enforce blind-watermarking-based IP protection, the impacts of robustness attackers must be properly addressed in further enhancements.

In addition, a combination of robustness attack and evasion attack (see Fig. 9) may exhaust the stego images prepared for ownership verification, significantly undermining the practicality and applicability of the blind-watermarking-based IP protection paradigm. To address this issue, it may be necessary to investigate a blind-watermarking-based IP protection technique that works equally well on new stego images generated by the model owner after the host model is trained, watermarked and shipped.

Table 5: Performance (classification accuracy) of the main task of the proposed E2E-Extraction and Two-Phase approaches under robustness attacks.

IP Protection Framework	Steganography-Host Pair	Gaussian Noise	Salt & Pepper Noise	Noiseless
Two-Phase Approach	En2D-109-ResNet18	64.8667%	61.9833%	68.9167%
	En2D-204-ResNet18	64.2333%	61.4167%	68.0333%
	En2D-1211-ResNet18	64.9000%	61.7667%	68.8833%
	En2D-2022-ResNet18	63.3167%	60.8000%	67.2167%
	GglNet-ResNet18	63.1000%	59.4667%	66.7000%
E2D-Extraction	En2D-2022-ResNet18	45.7833%	45.1167%	50.5667%

8 Conclusion and Future Work

In this paper, we re-examine the performance of blind-watermarking-based IP protection for DNN image classifiers on the more practical mini-ImageNet dataset and propose two enhanced IP protection techniques, which are evaluated from the security perspective. We find that existing blind-watermarking-based IP protection is still susceptible to various attacks, and the benefits of external ownership verification may be undermined or exploited. As our future work, we will further examine possible defenses against the attacks reported in this paper and further evaluate our proposed techniques in production systems.

Acknowledgments *Acknowledgments section is masked for double-blind review.*

References

1. ImageNet Classification. https://pjreddie.com/darknet/imagenet/
2. Models and Pre-Trained Weights – Torchvision 0.12 Documentations. https://pytorch.org/vision/stable/models.html (2017)
3. Aiken, W., Kim, H., Woo, S., Ryoo, J.: Neural Network Laundering: Removing Black-Box Backdoor Watermarks from Deep Neural Networks. Computers & Security **106**, 102277 (July 2021). https://doi.org/10.1016/j.cose.2021.102277, https://www.sciencedirect.com/science/article/pii/S0167404821001012
4. Baluja, S.: Hiding Images in Plain Sight: Deep Steganography. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 2066–2076. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (December 2017)
5. Baluja, S.: Hiding Images within Images. IEEE Transactions on Pattern Analysis and Machine Intelligence **42**(7), 1685–1697 (July 2020). https://doi.org/10.1109/TPAMI.2019.2901877
6. Batina, L., Bhasin, S., Jap, D., Picek, S.: CSI NN: Reverse Engineering of Neural Network Architectures through Electromagnetic Side Channel. In: 28th USENIX Security Symposium (USENIX Security 19). pp. 515–532. USENIX Association, Santa Clara, CA (August 2019), https://www.usenix.org/conference/usenixsecurity19/presentation/batina
7. Boroumand, M., Chen, M., Fridrich, J.: Deep residual network for steganalysis of digital images. IEEE Transactions on Information Forensics and Security **14**(5), 1181–1193 (May 2019). https://doi.org/10.1109/TIFS.2018.2871749
8. Cao, X., Jia, J., Gong, N.Z.: IPGuard: Protecting Intellectual Property of Deep Neural Networks via Fingerprinting the Classification Boundary. In: Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security. p. 14–25. ASIA CCS '21, Association for Computing Machinery, New York, NY, USA (May 2021). https://doi.org/10.1145/3433210.3437526, https://doi.org/10.1145/3433210.3437526
9. Chen, J., Wang, J., Peng, T., Sun, Y., Cheng, P., Ji, S., Ma, X., Li, B., Song, D.: Copy, Right? A Testing Framework for Copyright Protection of Deep Learning Models. cs.CR **abs/2112.05588** (December 2021). https://doi.org/10.48550/ARXIV.2112.05588

10. Chen, K., Guo, S., Zhang, T., Xie, X., Liu, Y.: Stealing Deep Reinforcement Learning Models for Fun and Profit. In: Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security. p. 307–319. ASIA CCS '21, Association for Computing Machinery, New York, NY, USA (May 2021). https://doi.org/10.1145/3433210.3453090
11. Chen, X., Wang, X., Zhang, K., Fung, K.M., Thai, T.C., Moore, K., Mannel, R.S., Liu, H., Zheng, B., Qiu, Y.: Recent Advances and Clinical Applications of Deep Learning in Medical Image Analysis. Medical Image Analysis **In Press**, 102444 (April 2022). https://doi.org/10.1016/j.media.2022.102444
12. Cheng, J., Tian, S., Yu, L., Gao, C., Kang, X., Ma, X., Wu, W., Liu, S., Lu, H.: ResGANet: Residual Group Attention Network for Medical Image Classification and Segmentation. Medical Image Analysis **76**, 102313 (February 2022). https://doi.org/10.1016/j.media.2021.102313
13. Fan, L., Ng, K.W., Chan, C.S., Yang, Q.: DeepIP: Deep Neural Network Intellectual Property Protection with Passports. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–1 (June 2021). https://doi.org/10.1109/TPAMI.2021.3088846
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
15. Hilty, R., Hoffmann, J., Scheuerer, S.: Intellectual Property Justification for Artificial Intelligence. Artificial Intelligence and Intellectual Property (April 2021). https://doi.org/10.1093/oso/9780198870944.003.0004
16. Hu, X., Chu, L., Pei, J., Liu, W., Bian, J.: Model Complexity of Deep Learning: A Survey. Knowledge and Information Systems **63**(10), 2585–2619 (August 2021). https://doi.org/10.1007/s10115-021-01605-0
17. Jing, J., Deng, X., Xu, M., Wang, J., Guan, Z.: HiNet: Deep Image Hiding by Invertible Network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 4733–4742 (October 2021)
18. Krizhevsky, A.: Learning Multiple Layers of Features from Tiny Images. Tech. rep. (April 2009)
19. Li, Z., Hu, C., Zhang, Y., Guo, S.: How to Prove Your Model Belongs to You: A Blind-Watermark Based Framework to Protect Intellectual Property of DNN. In: Proceedings of the 35th Annual Computer Security Applications Conference. p. 126–137. ACSAC '19 (December 2019). https://doi.org/10.1145/3359789.3359801
20. Lin, N., Chen, X., Lu, H., Li, X.: Chaotic Weights: A Novel Approach to Protect Intellectual Property of Deep Neural Networks. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **40**(7), 1327–1339 (July 2021). https://doi.org/10.1109/TCAD.2020.3018403
21. Liu, Y.: Tools for mini-Imagenet Dataset. https://github.com/yaoyao-liu/mini-imagenet-tools (October 2020)
22. Luo, W., Huang, F., Huang, J.: Edge Adaptive Image Steganography Based on LSB Matching Revisited. IEEE Transactions on Information Forensics and Security **5**(2), 201–214 (June 2010). https://doi.org/10.1109/TIFS.2010.2041812
23. Ong, D.S., Chan, C.S., Ng, K.W., Fan, L., Yang, Q.: Protecting Intellectual Property of Generative Adversarial Networks from Ambiguity Attacks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3630–3639 (June 2021)
24. Rathi, P., Bhadauria, S., Rathi, S.: Watermarking of Deep Recurrent Neural Network Using Adversarial Examples to Protect Intellectual Prop-

erty. Applied Artificial Intelligence **36**(1), 2008613 (December 2022). https://doi.org/10.1080/08839514.2021.2008613

25. ur Rehman, A., Rahim, R., Nadeem, S., ul Hussain, S.: End-to-End Trained CNN Encoder-Decoder Networks for Image Steganography. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops (September 2018)
26. Rokhana, R., Herulambang, W., Indraswari, R.: Multi-Class Image Classification Based on MobileNetV2 for Detecting the Proper Use of Face Mask. In: 2021 International Electronics Symposium (IES). pp. 636–641 (September 2021). https://doi.org/10.1109/IES53407.2021.9594022
27. Rouhani, B.D., Chen, H., farinaz Koushanfar: DeepSigns: A Generic Watermarking Framework for Protecting the Ownership of Deep Learning Models. Cryptology ePrint Archive, Paper 2018/311 (June 2018), https://eprint.iacr.org/2018/311, https://eprint.iacr.org/2018/311
28. Subramanian, N., Elharrouss, O., Al-Maadeed, S., Bouridane, A.: Image Steganography: A Review of the Recent Advances. IEEE Access **9**, 23409–23423 (January 2021). https://doi.org/10.1109/ACCESS.2021.3053998
29. Tang, R., Du, M., Hu, X.: Deep Serial Number: Computational Watermarking for DNN Intellectual Property Protection. CoRR **abs/2011.08960** (November 2020), https://arxiv.org/abs/2011.08960
30. Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., Wierstra, D.: Matching Networks for One Shot Learning. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. p. 3637–3645. NIPS'16, Curran Associates Inc., Red Hook, NY, USA (2016), https://dl.acm.org/doi/10.5555/3157382.3157504
31. Wu, H., Liu, G., Yao, Y., Zhang, X.: Watermarking neural networks with watermarked images. IEEE Transactions on Circuits and Systems for Video Technology **31**(7), 2591–2601 (July 2021). https://doi.org/10.1109/TCSVT.2020.3030671
32. Xiang, Z., Sang, J., Zhang, Q., Cai, B., Xia, X., Wu, W.: A New Convolutional Neural Network-Based Steganalysis Method for Content-Adaptive Image Steganography in the Spatial Domain. IEEE Access **8**, 47013–47020 (March 2020). https://doi.org/10.1109/ACCESS.2020.2978110
33. Xu, G., Wu, H.Z., Shi, Y.Q.: Ensemble of CNNs for Steganalysis: An Empirical Study. In: Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security. p. 103–107. IH&MMSec '16, Association for Computing Machinery, New York, NY, USA (June 2016). https://doi.org/10.1145/2909827.2930798
34. Xu, G., Wu, H.Z., Shi, Y.Q.: Structural Design of Convolutional Neural Networks for Steganalysis. IEEE Signal Processing Letters **23**(5), 708–712 (May 2016). https://doi.org/10.1109/LSP.2016.2548421
35. Xue, M., Zhang, Y., Wang, J., Liu, W.: Intellectual Property Protection for Deep Learning Models: Taxonomy, Methods, Attacks, and Evaluations. IEEE Transactions on Artificial Intelligence **1**(01), 1–1 (August 2022). https://doi.org/10.1109/TAI.2021.3133824
36. Ye, J., Ni, J., Yi, Y.: Deep Learning Hierarchical Representations for Image Steganalysis. IEEE Transactions on Information Forensics and Security **12**(11), 2545–2557 (November 2017). https://doi.org/10.1109/TIFS.2017.2710946
37. Zhang, J., Gu, Z., Jang, J., Wu, H., Stoecklin, M.P., Huang, H., Molloy, I.: Protecting Intellectual Property of Deep Neural Networks with Watermarking. In: Proceedings of the 2018 on Asia Conference on Computer and Communications Security. p. 159–172. ASIACCS '18, Association for Computing Machinery, New York, NY, USA (May 2018). https://doi.org/10.1145/3196494.3196550

38. Zhang, J., Chen, D., Liao, J., Zhang, W., Feng, H., Hua, G., Yu, N.: Deep Model Intellectual Property Protection via Deep Watermarking. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–1 (March 2021). https://doi.org/10.1109/TPAMI.2021.3064850
39. Zheng, L.: How to Prove Your Model Belongs to You: A Blind-Watermark Based Framework to Protect Intellectual Property of DNN. https://github.com/zhenglisec/Blind-Watermark-for-DNN (January 2021)