Adversarial Attack for Robust Watermark Protection Against Inpainting-based and Blind Watermark Removers

Mingzhi Lyu*
lyum0002@e.ntu.edu.sg
Nanyang Technological University
Singapore, Singapore

Yi Huang*
yi.huangy@ntu.edu.sg
Nanyang Technological University
Singapore, Singapore

Adams Wai-Kin Kong
adamskong@ntu.edu.sg
Nanyang Technological University
Singapore, Singapore

ABSTRACT

The rise of social media platforms, especially those focusing on image sharing, has made visible watermarks increasingly important in protecting image copyrights. However, multiple studies have revealed that watermarks are vulnerable to both inpainting-based removers and blind watermark removers. Though two adversarial attack methods have been proposed to defend against watermark removers, they are tailored to a particular type of removers in a white-box setting, which significantly limits their practicality and applicability. To date, there is no adversarial attack method that can protect watermarks against the two types of watermark removers simultaneously. In this paper, we propose a novel method, named Adversarial Watermark Defender with Attribution-Guided Perturbation (AWD-AGP), that defends against both inpainting-based and blind watermark removers under a black-box setting. AWD-AGP is the first watermark protection method employing adversarial location. The adversarial location is generated by a Watermark Positioning Network, which predicts an optimal location for watermark placement, making watermark removal challenging for inpainting-based removers. Since inpainting-based removers and blind watermark removers exploit information in different regions of an image to perform removal, we propose an attribution-guided scheme, which automatically assigns attack strengths to different pixels against different removers. With this design, the generated perturbation can attack the two types of watermark removers concurrently. Experiments on seven models, including four inpainting-based removers and three blind watermark removers demonstrate the effectiveness of AWD-AGP.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision tasks**; • **Security and privacy** → **Privacy protections**.

KEYWORDS

Watermark; inpainting-based watermark remover; blind watermark remover; adversarial attack

*Both authors contributed equally to this research.

MM '23, October 29-November 3, 2023, Ottawa, ON, Canada

ACM ISBN 979-8-4007-0108-5/23/10.
https://doi.org/10.1145/3581783.3612034

ACM Reference Format:
Mingzhi Lyu, Yi Huang, and Adams Wai-Kin Kong. 2023. Adversarial Attack for Robust Watermark Protection Against Inpainting-based and Blind Watermark Removers. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29-November 3, 2023, Ottawa, ON, Canada.* ACM, Ottawa, Ontario, Canada, 10 pages. https://doi.org/10.1145/3581783.3612034

1 INTRODUCTION

With the increasing popularity of social media for image sharing such as Instagram, Tumblr, and Flickr, visible watermarks have become an important tool for protecting image copyright. Typically, such watermarks are created by superimposing a partially transparent image, which may include a company name or a logo, onto an original source image as depicted in Fig. 1 (a) and (d). However, numerous studies have demonstrated that visible watermarks can be easily removed by advanced deep-learning models, including inpainting models and blind watermark removers. Inpainting models [15, 12, 35, 36, 23] are designed to reconstruct missing regions in an image, making them applicable to removing visible watermarks. Adversaries only need to crop out the watermark region and use inpainting models to reconstruct the cropped region. These models can infer missing contents based on the information present in the rest of the image, and they are capable of producing high-quality results that appear natural and realistic to the human eye (as illustrated in Fig. 1 (b)). In the paper, watermark removal methods based on inpainting models are referred to as inpainting-based watermark removers. Though inpainting-based watermark removers are capable of generating high-fidelity watermark-free images, they require prior knowledge of the watermark location, which prevents them from removing watermarks automatically. To explore the feasibility of fully automatic watermark removal, numerous blind watermark removers that do not require prior knowledge of the watermark have been developed in recent years [2, 16, 5, 20, 17]. Blind watermark removers can be used without any human intervention, making them suitable for automatically removing visible watermarks in batches. Fig. 1 (e) is a resultant image from a blind watermark remover.

As watermark removal technologies continue to advance, traditional watermarking methods are becoming less effective in protecting image copyrights. Therefore, there is an urgent need to develop powerful and reliable protection methods to safeguard against these watermark removers. However, only limited work has been developed to defend against deep learning based watermark removal methods. Recently, Khachaturov et al. [14] introduced an attack named markpainting to protect watermarked images against inpainting-based watermark removers with adversarial perturbations. Liu et al. [19] proposed Watermark Vaccine (WV) as

Figure 1: Watermarked images and the removal results. (a) and (d) are original watermarked images. (b) and (c) are respectively watermark removal results from an inpainting model before and after WV protection [19]. (e) and (f) are respectively watermark removal results from a blind watermark remover before and after markpainting [14].

a proactive defense against blind watermark removal networks. Although markpainting and WV perform well in a white-box setting, they suffer from low transferability to other inpainting-based and blind watermark removers. This limits their practical use as image owners are unlikely to have perfect knowledge of the specific watermark removal models utilized by potential adversaries. Additionally, they consider either inpainting-based or blind watermark removers, making their protection ineffective against the other type of watermark removers (as shown in Fig. 1 (c) and (f)). In reality, adversaries may try blind watermark removers first and then switch to inpainting-based removers if the blind removers fails. To address these challenges, we propose a novel method, named Adversarial Watermark Defender with Attribution-Guided Perturbation (AWD-AGP), which is designed to prevent the original content covered by watermarks from being restored by both types of watermark removal techniques while removing the watermarks.

The proposed protection method, AWD-AGP consists of two key elements: adversarial location for watermark embedding and attribution-guided adversarial perturbations generation. We notice that the position of a watermark plays a critical role in the effectiveness of inpainting-based watermark removal techniques. Watermarks placed in regions with different semantic information from surrounding pixels are particularly challenging to these techniques. For instance, if a partially transparent watermark completely covers an object, it may not be possible for inpainting models to guess the object without any prior information. Building on this observation, we introduce a Watermark Positioning Network that can predict an adversarial location for watermark placement with limited semantic connections to the surrounding pixels. By placing a watermark in this specific location, it becomes more challenging for all inpainting-based watermark removers to restore the original contents. [1] Furthermore, we observe that the attribution maps of inpainting-based removers and blind watermark removers often highlight different regions of an image. It indicates that they use information in different regions to perform removal. In light of this, we propose an attribution-guided perturbation scheme to generate adversarial perturbations that can attack both inpainting-based and blind watermark removers. Specifically, our scheme utilizes the attribution maps of both types of removers to guide adversarial perturbations in different regions to attack different types of removers, so that the effectiveness of each region of perturbation can be maximized. Moreover, since the same type of removers shares similar attribution maps, the attribution-guided perturbation scheme can also enhance the transferability of the generated adversarial perturbations among the same type of watermark removers. By adding these attributed-guided adversarial perturbations into images with watermarks positioned according to the Watermark Positioning Network, we achieve robust protection against both types of watermark removers.

To validate the effectiveness of the proposed method, we conduct the evaluation on three blind watermark removers, including WDModel [20], DBWEModel [5] and SLBRModel [17], and four inpainting models, including Gennet [35], Crfillnet [37], FcFnet [12] and Matnet [15]. Our experimental results demonstrate the effectiveness of our proposed method against both inpainting-based and blind watermark removers. Additionally, we conduct a comprehensive ablation study to confirm the effectiveness of each proposed technique in our method.

2 RELATED WORK

Visible Watermark Removal Visible watermark removal has been an active research area for several years. Early attempts involved using traditional image processing techniques such as filtering and thresholding. However, due to their limited efficiency and effectiveness, these methods pose a lower threat to copyright protection based on visible watermarks and are not included in our study. The advancement of generative models has introduced new possibilities for watermark removal. An adversary can remove watermarks simply by cropping out the region with unwanted watermarks and filling in the missing information with advanced inpainting models [15, 12, 35, 37]. Then, the adversary can get an image that looks natural and is free of any watermarks. This approach requires users to identify the watermark location, which limits its application in removing watermarks in batches. Some recent blind watermark removers [2, 16, 5, 20, 17] attempt to tackle the problem of blind watermark removal using deep learning techniques in an end-to-end manner. Some works [2, 16] formulate it as an image-to-image translation task where generative adversarial networks were utilized to map watermarked images to watermark-free ones, without the need to localize the watermark. However, the variability in the size, shape, color, and transparency of watermarks poses a significant challenge to these methods. To address these issues, some studies [5, 20, 17] propose to utilize two-stage strategies for achieving better performance. The primary goal of the first stage is to generate a mask for localizing the watermarks and to perform a

[1] The opacity of a watermark is set to less than 40% to avoid the watermark affecting the original content.

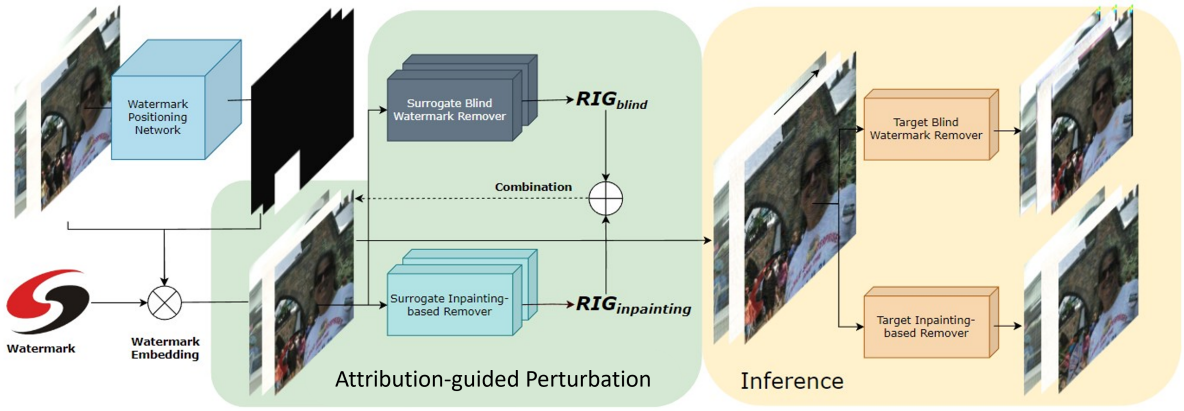


Figure 2: An overview of the proposed method. Original images will be firstly input to the watermark positioning network to obtain masks with adversarial effect for watermark embedding. Next, watermarks will be attached to the images at the predicted masks, and surrogate inpainting-based and blind watermark removers will be utilized to generate adversarial perturbation for the watermarked images. During perturbation generation, attribution maps of watermark removers obtained by RIG[9] will guide the perturbations in different regions to attack different types of watermark removers.

coarse restoration, whereas the second stage concentrates specifically on the watermarked region to enhance the removal results further. Compared to blind watermark removers without localizing the watermarks, these methods achieve much better removal performance. Therefore, for the blind watermark removers, we choose the two-stage removers [5, 20, 17] as our target models.

Adversarial Attacks for Protection After Szegedy et al. first revealed the vulnerability of deep neural networks to adversarial examples in 2013 [30], there has been a significant amount of research on adversarial attacks across various tasks, such as classification [24, 3, 7] and detection [31, 4, 11, 10], as well as segmentation [1, 32]. These studies aim to investigate the vulnerability of target models, which in turn helps to develop more reliable models for real-world applications. In recent years, some works have proposed the use of adversarial perturbations to protect images from being manipulated by deep generative models [27, 34, 8, 28, 21, 19]. For instance, Ruiz et al. [27] and Huang et al. [8] suggest using adversarial examples to defend face images against face attribution manipulation networks. Sun et al. [28] and Anej et al. [34] propose attacks against face swapping models to stop them from generating valid face images. Recently, Khachaturov et al. [14] propose a method called markpainting to prevent watermark removal by inpainting-based watermark removers. It forces a target inpainting-based watermark remover to generate the original content in the specified mask region, thus making the image resistant to it. However, their protection against unseen inpainting-based removers is not satisfactory as targeted adversarial examples are difficult to transfer [21]. Liu et al. [19] propose Disrupting Watermark Vaccine (DWV) and Inerasable Watermark Vaccine (IWV) to protect watermarked images against blind watermark removers. DWV is designed to cause damage to the host image after it passes through watermark-removal networks. On the other hand, IWV aims to keep the watermark intact after the watermarked image passes through blind watermark-removal networks. However, the authors report that both IWV and DWV show limited transferability across unseen blind watermark removers. To summarize, existing methods, including markpainting, DWV, and IWV are only capable of safeguarding watermarked images against a specific type of watermark removal techniques in a white-box setting.

3 METHOD

Our method is designed to provide protection for watermarked images against both inpainting-based and blind watermark removers. It consists of two main components, adversarial location for watermark embedding and attribution-guided adversarial perturbations generation. Inpainting-based removers first erase the image content and watermark within a user provided mask and refill the image content based on the surrounding information, making their performance highly sensitive to the location of the watermark. Thus, the proposed method first identifies an adversarial position for inserting a watermark, which can degrade the effectiveness of inpainting-based removers. In this study, we observe that the attribution maps of inpainting-based removers and blind watermark removers highlight different regions of an image, implying that they rely on information in different regions to perform the removal. To leverage this difference, we propose an attribution-guided scheme to automatically allocate attack strength to different pixels for different types of removers. With this scheme, we are able to generate perturbations that can effectively defend both inpainting-based removers and blind watermark removers. Fig. 2 is an overview of our proposed method.

3.1 Watermark Positioning Network

Inpainting-based removers analyze the surrounding pixels of the user provided mask and use this information to infer the content of the missing pixels. However, when the original content within the mask has limited semantic connection with the surrounding pixels, inpainting-based removers may struggle to accurately predict the original content, leading to a degraded output. Therefore, we propose a Watermark Positioning Network (WPN) which predicts

(a) (b) (c)

Figure 3: An example of the original superpixels (a) and their superpixel clusters (b). In (c), superpixel clusters completely covered by the mask are highlighted in yellow and the mask is highlighted in green.

an adversarial position for watermark embedding to prevent the removers from accurately recovering the original content within the mask. Except for the position of embedding, the size of a watermark also affects the performance of inpainting-based watermark removers. When a watermark is very large, covering the entire image, inpainting-based removers are unable to function effectively as they have to remove all of the image content before refilling it. To eliminate this extreme case and only focus on position, we assume that a user has a predefined mask size for watermark embedding and seeks recommendations for the optimal location to embed it. In this study, the mask size is set to $\frac{1}{3}$ of the image height by $\frac{1}{3}$ of the image weight. To assess the semantic connections between pixels, we employ SpixelFCN [33] to segment an image into superpixels, which group pixels with similar semantic meanings together. Furthermore, we perform post-processing on the superpixels by merging adjacent superpixels with similar semantic content into superpixel clusters, thereby reducing semantic similarity between adjacent superpixels. Further details regarding the post-processing step can be found in the appendix. Fig. 3 shows the original superpixels and their superpixel clusters. Isolated score S based on the superpixel clusters is then used as a metric for selecting an adversarial mask region. It is defined as the ratio of the total area of superpixel clusters that are completely covered by the mask to the size of the mask (demonstrated in Fig. 3 (c)).

Due to the discrete nature of the mask, i.e., inside or outside the mask region, differential evolution (DE), a gradient-free optimization algorithm, can be employed to identify the optimal mask with the highest S. However, DE is extremely slow in locating an adversarial position, which severely restricts its application in scenarios with limited computational resources or high throughput requirements. Although a network can be trained to speed up the inferences, it requires running DE numerous times to generate enough optimal mask positions as training data. In other words, significant computational costs are required to generate enough samples for training the network. DE seeks the optimal position by iterating multiple times on one image, which progressively generates a mask with a higher S in each iteration. If the data from the intermediate iterations can be used, the time and computation cost to generate the training data could be significantly reduced. In light of this, we propose a new training scheme, which not only uses the final predictions but also the intermediate results of DE to train the network. Specifically, the intermediate masks of an image produced by DE are arranged in increasing order according to their isolated score $\boldsymbol{S}_{DE_j}$ to form a set of ground truth masks for that image. During training, a mask m_j from this set is sequentially selected as the ground truth position for that image in each iteration. By training in this manner, the network is taught to progressively predict a mask with higher $\boldsymbol{S}_{DE_j}$, which reduces the time and computation cost to generate the training data.

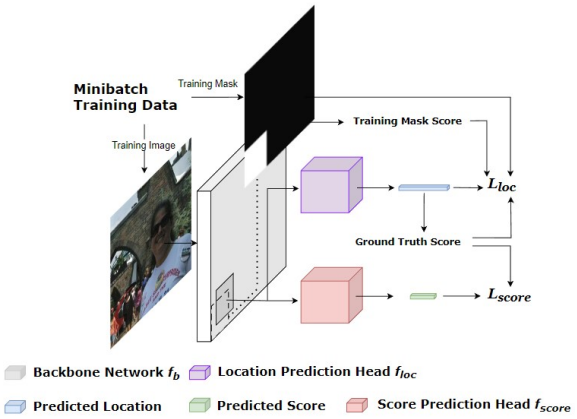


Figure 4: An illustration of the Watermark Positioning Network.

WPN predicts both isolated score S and a corresponding mask location, which shares some similarities with object detection networks. Thus, WPN adopts the design of Region Proposal Network (RPN) from FasterRCNN [25], which includes a backbone, a location prediction head, and a score prediction head. Fig. 4 is an illustration of WPN. The backbone network f_b extracts features from input images, where each pixel in the feature map corresponds to a mask proposal with the predefined mask size. The location prediction head f_{loc} calculates the offset coordinates with respect to each proposal position, while the score prediction head f_{score} calculates the isolated score for each proposal. The objective function is defined as follows:

$$L = L_{loc}(\boldsymbol{loc}_p, loc_{m_j}, S_{gt\_m_j}) + L_{socre}(\boldsymbol{S}_{pred\_p}), \tag{1}$$

where $\boldsymbol{loc}_p$ are the offsets predictions for the mask proposals of an input image $\boldsymbol{x}$. loc_{m_j} is the corresponding ground truth mask m_j in the j^{th} iteration, and $\boldsymbol{S}_{gt\_m_j}$ is the isolated score of m_j. $\boldsymbol{S}_{pred\_p}$ are the predicted scores for the mask proposals. L_{loc} is a modified of the SmoothL1Loss and it is defined as follows:

$$\begin{aligned} L_{loc}(\boldsymbol{loc}_p, loc_{m_j}, S_{gt\_m_j}) = \sum_i ((\frac{\sigma}{2}\left|loc_{p_i} - loc_{m_j}\right|^2) \times \mathbb{1}_{l1} \\ + (\left|loc_{p_i} - loc_{m_j}\right| - \frac{\sigma}{2}) \times (1 - \mathbb{1}_{l1})) \times \mathbb{1}_{overlap}, \end{aligned} \tag{2}$$

where loc_{p_i} is the offset prediction for the i^{th} mask proposal p_i. $\mathbb{1}_{l1}$ an indicator function equals to 1 when $\left|loc_{p_i} - loc_{m_j}\right| < \frac{1}{\sigma}$; otherwise it equals to 0. σ is a hyperparameter to control the effect of the two terms in equation (2). $\mathbb{1}_{overlap}$ an indicator function equals to 1 when the following two conditions are satisfied: 1) the Intersection over Union (IoU) between a mask proposal and the ground truth mask m_j for that iteration exceeds a threshold T and

2) $S_{gt\_p_i}$ is smaller than $S_{gt\_m_j}$. $S_{gt\_p_i}$ is the ground truth isolated score of the proposal loc_{p_i}, which is directly calculated as the ratio of the total area of superpixel clusters completely covered by the proposal to the size of the proposal. Otherwise, $\mathbb{1}_{overlap}$ equals to 0. The aim of $\mathbb{1}_{overlap}$ is to only penalize the low quality proposals which have lower isolated scores than the current ground truth m_j. As a result, the network will be encouraged to propose better masks while avoiding being deteriorated by low quality training data. L_{socre} is defined as follow:

$$L_{score}(S_{pred\_p}) = \frac{1}{n}\sum_{i=0}^{n}(S_{gt\_p_i} - S_{pred\_p_i})^2, \tag{3}$$

where $S_{pred\_p_i}$ is the predicted score for the i^{th} mask proposal p_i and n is the total number of proposals. Note that in each iteration, L_{score} incentivizes all proposals to predict their accurate isolated scores S, whereas L_{loc} only encourages the network to output a location close to the ground truth mask in that iteration. In other words, minimizing L_{loc} motivates the network to generate masks with higher scores, while minimizing L_{score} encourages the network to accurately predict the scores for each mask proposal.

3.2 Attribution-guided Adversarial Perturbations

Given a watermarked image, we aim to generate adversarial perturbations which can fool both inpainting-based and blind watermark removers. A possible solution is to sum the adversarial perturbations generated by both types of removers in each pixel. However, this approach assumes that pixels in different regions are equally important to different types of removers. This assumption does not align with their attribution maps shown in Fig. 5, which reveal that the mask regions are more critical to blind watermark removers, whereas the adjacent pixels surrounding the mask regions are more important to inpainting-based removers. Therefore, we propose attribution-guided adversarial perturbations to automatically allocate different attack strengths to each pixel based on the attribution maps of different removers. The proposed attack utilizes Random path Integrated gradients (RIG), which is a variant of Integrated gradients (IG) [29] to compute the attribution maps for different types of watermark removers. IG is an attribution method designed based on two fundamental axioms - sensitivity and implementation invariance. These axioms ensure that IG provides a reliable and accurate approach to analyse the behavior of deep neural networks. Additionally, IG satisfies the axiom of completeness, which plays an important role in the design of the proposed attack. To explain our attack, we first briefly introduce IG and RIG below.

IG is computed by integrating the gradients of the output prediction with respect to the input feature along a straight-line path from a reference image $\boldsymbol{r}$ to the input image $\boldsymbol{x}$. The integrated gradient of the i^{th} pixel of input $\boldsymbol{x}$ is computed by the following equation:

$$\mathrm{IG}_i(f, \boldsymbol{x}, \boldsymbol{r}) = (x_i - r_i)\int_{\alpha=0}^{1}\frac{\partial f(\boldsymbol{r} + \alpha \times (\boldsymbol{x} - \boldsymbol{r}))}{\partial x_i}\, d\alpha, \tag{4}$$

where r_i is the i^{th} pixel of $\boldsymbol{r}$, x_i is the i^{th} pixel of $\boldsymbol{x}$, and f is a deep network $f : \mathbb{R}^n \rightarrow \mathbb{R}$. RIG [9] is a variant of IG that computes the integral along a random piecewise linear path. RIG also satisfies the axiom mentioned above and it includes an additional augmentation step that results in better smoothing [9]. Therefore, we employ RIG in our attack. It is calculated by the following equation:

$$\mathrm{RIG}_i(f, \boldsymbol{x}, \boldsymbol{r}) = \sum_{e=0}^{E-1}\mathrm{IG}_i(f, \boldsymbol{x}_{e+1}, \boldsymbol{x}_e), \tag{5}$$

where $\boldsymbol{x}_0, \cdots, \boldsymbol{x}_E$ are the $E+1$ turning points of a random piecewise linear path with $\boldsymbol{x}_0$ and $\boldsymbol{x}_E$ being the starting point and end point respectively. The turning points $\boldsymbol{x}_e$ are generated by:

$$\boldsymbol{x}_e = \boldsymbol{x}_0 + \frac{e}{E}(\boldsymbol{x}_E - \boldsymbol{x}_0) + \boldsymbol{v}, \tag{6}$$

where $\boldsymbol{v}$ is a random vector following a uniform distribution with support from $(-\tau, \tau)$.

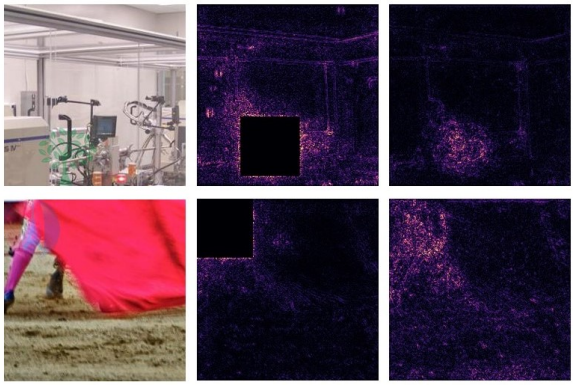


Figure 5: The attribution map of an ensemble of inpainting-based removers $f_{in_{en}}$ and an ensemble of blind watermark removers $f_{bwr_{en}}$. The images from the first to the third column are the original images, the attribution maps from $f_{in_{en}}$ and $f_{bwr_{en}}$, respectively.

To generate the perturbations, VGG perceptual loss [13] is taken as a dissimilarity metric. Our aim is to maximize the perceptual difference between the original outputs and attacked outputs of different watermark removers. The objective function can be written as follow:

$$\begin{aligned} &\max_{\boldsymbol{\delta}} L_{ppl}(f_{en}(\boldsymbol{x_w}), f_{en}(\boldsymbol{x_w} + \boldsymbol{\delta})) \\ &s.t. \quad \|\boldsymbol{\delta}\|_\infty < \epsilon, \end{aligned} \tag{7}$$

where $\boldsymbol{x_w}$ is a watermarked image, L_{ppl} is the perceptual loss, and f_{en} serving as a surrogate model is an ensemble of inpainting-based removers and blind watermark removers. $\boldsymbol{\delta}$ is the adversarial perturbations and ϵ is the attack budget defined by L_∞ norm. We calculate the RIG of different models based on the above objective function. Thus, the calculated RIG is the attention score of each pixel to the target loss function and the adversarial perturbations can be updated based on the following equation:

$$\begin{aligned} \boldsymbol{x}_{\boldsymbol{w}}^{'} = \boldsymbol{x_w} - \alpha \times \mathrm{sign}\Bigg(& N_{l_1}\Bigg(\sum_{i=1}^{k_1} N_{l_1}\left(\mathrm{RIG}\left(f_{in_i}, \boldsymbol{x_w}, \boldsymbol{r}\right)\right)\Bigg) \\ & + N_{l_1}\Bigg(\sum_{i=1}^{k_2} N_{l_1}\left(\mathrm{RIG}\left(f_{bwr_i}, \boldsymbol{x_w}, \boldsymbol{r}\right)\right)\Bigg)\Bigg), \end{aligned} \tag{8}$$

Table 1: The attack results of AWD-AGP and the baselines. The metrics are calculated between the mask region of the original output and that of the attacked output of target models. Note that the results of AGP and ensemble are discussed in Section 4.3

Target Model	Surrogate Models	Postion	Method	SSIM ↓	PSNR ↓	L_{ppl} ↑	RMSE ↑	REMSE ↑
WDModel	Crfillnet Gennet	Random	Markpainting	0.977	33.906	0.097	0.025	0.173
	SLBRModel DBWEModel	Random	DWV	0.919	29.029	0.344	0.050	0.238
	Crfillnet Gennet SLBRModel DBWEModel	Random	AGP	0.856	25.345	0.654	0.079	0.279
	Crfillnet Gennet SLBRModel DBWEModel	WPN	Ensemble	0.881	25.950	0.585	0.068	0.279
	Crfillnet Gennet SLBRModel DBWEModel	WPN	AWD-AGP	**0.846**	**24.210**	**0.771**	**0.083**	**0.301**
Matnet	Crfillnet Gennet	Random	Markpainting	0.682	19.806	1.063	0.128	0.335
	SLBRModel DBWEModel	Random	DWV	0.733	22.637	0.806	0.100	0.324
	Crfillnet Gennet SLBRModel DBWEModel	Random	AGP	0.670	20.781	1.072	0.117	0.347
	Crfillnet Gennet SLBRModel DBWEModel	WPN	Ensemble	0.624	18.166	**1.307**	0.147	0.363
	Crfillnet Gennet SLBRModel DBWEModel	WPN	AWD-AGP	**0.604**	**17.820**	1.302	**0.153**	**0.372**
FcFnet	Crfillnet Gennet	Random	Markpainting	0.665	16.673	1.249	**0.177**	0.338
	SLBRModel DBWEModel	Random	DWV	0.863	27.053	0.465	0.058	0.291
	Crfillnet Gennet SLBRModel DBWEModel	Random	AGP	0.677	18.324	1.193	0.165	0.346
	Crfillnet Gennet SLBRModel DBWEModel	WPN	Ensemble	0.683	18.104	1.146	0.150	0.354
	Crfillnet Gennet SLBRModel DBWEModel	WPN	AWD-AGP	**0.635**	**16.511**	**1.316**	**0.177**	**0.367**

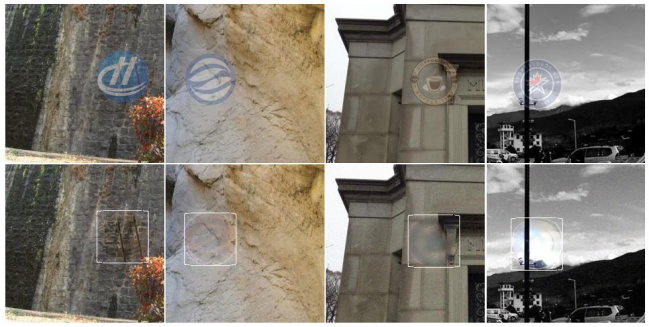
Table 2: The restoration and attack results for watermarks in different regions on inpainting models. WPN is the adversarial mask region generated by WPN. The metrics under the restoration without attack are calculated from the mask regions between the original watermark free images and the restoration outputs generated by the inpainting models. The metrics for ensemble attacks are calculated between the mask region of the original output and that of the restoration output generated by inpainting models.

		Restoration without Attack					Ensemble Attack				
Target Model	Position	SSIM ↓	PSNR ↓	L_{ppl} ↑	RMSE ↑	REMSE ↑	SSIM ↓	PSNR ↓	L_{ppl} ↑	RMSE ↑	REMSE ↑
Crfillnet	Center	0.677	20.231	0.972	0.139	0.318	0.790	23.796	0.715	0.078	0.303
	Left Top	0.640	18.448	1.128	0.178	0.317	0.784	23.604	0.778	0.081	0.313
	Random	0.676	20.537	0.996	0.145	0.318	0.784	23.529	0.756	0.081	0.307
	WPN	**0.535**	**14.594**	**1.521**	**0.219**	**0.360**	**0.757**	**22.471**	**0.898**	**0.088**	**0.343**
Gennet	Center	0.568	17.470	1.179	0.175	0.326	0.600	18.721	1.390	0.142	0.359
	Left Top	0.561	16.505	1.270	0.206	0.321	0.585	18.367	1.494	0.153	0.366
	Random	0.567	17.790	1.189	0.177	0.323	0.593	18.291	1.436	0.151	0.359
	WPN	**0.454**	**12.979**	**1.708**	**0.258**	**0.359**	**0.554**	**16.669**	**1.639**	**0.171**	**0.382**
FcFnet	Center	0.676	20.314	0.991	0.142	0.323	0.737	20.451	0.945	0.130	0.328
	Left Top	0.654	18.534	1.046	0.184	0.312	0.718	19.563	1.000	0.143	0.325
	Random	0.676	20.527	1.008	0.148	0.322	0.735	20.614	0.968	0.130	0.328
	WPN	**0.540**	**14.368**	**1.491**	**0.227**	**0.358**	**0.683**	**18.104**	**1.146**	**0.150**	**0.354**
Matnet	Center	0.687	20.403	0.946	0.141	0.314	0.682	20.825	1.098	0.114	0.344
	Left Top	0.669	18.627	0.988	0.182	0.301	0.667	19.730	1.162	0.137	0.336
	Random	0.691	20.771	0.949	0.143	0.312	0.682	20.967	1.103	0.116	0.344
	WPN	**0.559**	**14.547**	**1.415**	**0.223**	**0.347**	**0.624**	**18.166**	**1.307**	**0.147**	**0.363**

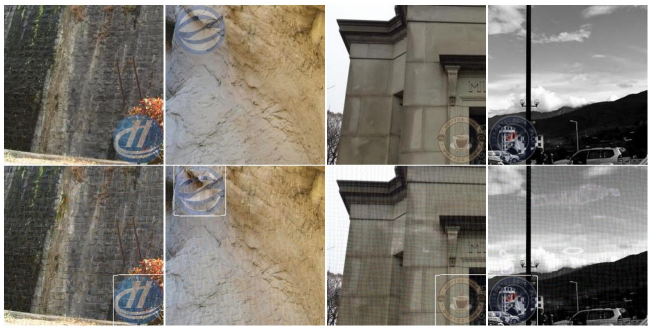
where f_{in_i} and f_{bwr_i} are the i^{th} inpainting-based and blind watermark remover used in training respectively. k_1 and k_2 are the numbers of inpainting models and blind watermark removers respectively. N_{l1} is the $l1$ normalization process. Before combining RIG, the RIG values are first normalized within each type of remover to make each of them contribute equally to the final attack. Thanks to the axiom of *Completeness*, the above equation can be used to perform white-box attacks and enhance the transferability in black-box attacks [9]. By updating in this way, pixels with higher RIG values for one type of remover can contribute more in attacking that type of remover.

4 EXPERIMENTS

In this section, we first introduce our experimental settings and some implementation details. We then compare the performance of AWD-AGP with the state-of-the-art adversarial attacks for watermark protection, including markpainting [14] and DWV [19], against both types of watermark removers. To further evaluate the efficacy of AWD-AGP, we conduct extensive ablation studies to analyze the impact of each component of the proposed method.

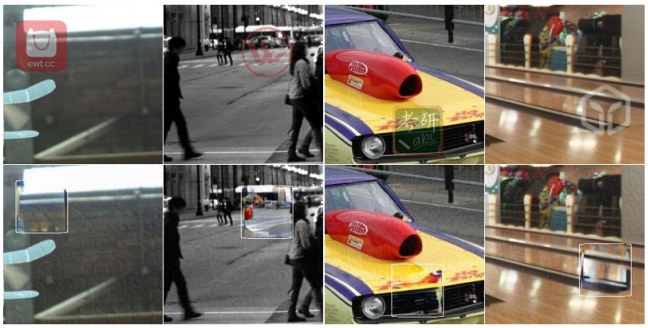


(a) DWV



(b) AWD-AGP

Figure 6: The attack results of AWD-AGP and DWV on WD-Model. The images in the first row are watermarked images and the images in the second row are the attack results.



(a) Markpainting

(b) AWD-AGP

Figure 7: The attack results of AWD-AGP and markpainting on FcFnet. The images in the first row are watermarked images and the images in the second row are the attack results.

4.1 Experimental Setups

Model and Dataset. There are seven models involved in the evaluation, which include four inpainting-based removers (Gennet [35], Crfillnet [37], FcFnet [12], Matnet [15]) and three blind watermark removers (WDModel [20], DBWEModel [5] SLBRModel [17]). The inpainting-based removers are trained on the Place2 dataset, while the blind watermark removers are trained on either COCO [18] or PASCAL VOC2012 [6] dataset. To evaluate AWD-AGP on them simultaneously, the three blind watermark removers are fine-tuned on a dataset consisting of 20,000 watermark-free images randomly sampled from the training set of Place2 and 20,000 watermarked images that are created by randomly embedding the watermarks from CLWD [20] onto the 20,000 watermark-free images. The four inpainting-based removers used in the evaluation are pre-trained models provided by the authors. The links to these models can be found in the appendix. Another 500 images sampled from the validation set of Place2 are taken as watermark-free images, and all the watermarked images used in the evaluation are generated by embedding the watermarks in CLWD into different positions of the 500 images.

Metrics and Parameters. Following previous works [14, 19], VGG perceptual loss (L_{ppl}), structural index similarity (SSIM), peak signal-to-noise ratio (PSNR), and root-mean-square error (RMSE) are employed as the evaluation metrics to evaluate the attack performance. We further include edge root mean square (ERMSE) to measure the edge changes before and after attacks. It is computed as the root mean square error between the edges of two images, where the edges are detected by the Canny edge detector with its default parameters [26]. The attack budget ϵ is set to 0.03. The step size α is set to 0.02. 100 iterations are used to generate the perturbations for baselines and 30 iterations are used for AWD-AGP. The RIG is computed with the 30 turning points and τ is set to 0.03. T is set to 0.7 and σ in equation (2) is set to 3.

4.2 Effectiveness of AWD-AGP

In this section, we compare AWD-AGP with markpainting and DWV under the same $\epsilon = 0.03$ on different types of watermark removers. For DWV, we use the default setting, except for ϵ. Since IWV is designed for targeted attacks by modifying the objective function of DWV, while AWD-AGP is designed for untargeted attacks, it is inappropriate to directly compare the two attacks as they have different goals. Thus, we do not include IWV in our evaluation. Markpainting is also designed for target attack, whose objective is to make the inpainting output identical to the watermarked image. As the authors of markpainting do not release their code, we re-implement markpainting based on their paper. However, we find that the transferability of markpainting is very low in a black-box setting, which aligns with the findings reported in their original paper. To ensure a fair comparison, we slightly modify the objective function of markpainting by changing it to an untargeted attack to increase its transferability across other models. As AWD-AGP

employs both inpainting models and blind watermark removers as surrogate models, we randomly select two inpainting models and two blind watermark removers as surrogate models, while leaving one blind watermark remover and two inpainting models as target models. We sample the surrogate models twice to better evaluate the effectiveness of AWD-AGP. Due to the space limit, we present the experimental results from one sample of the surrogate models in Table 1. The other is given in the appendix. Since the baseline methods are designed for only one type of removers, we use the corresponding type of removers in the sample as their surrogate models. For example, the two inpainting surrogate models used by AWD-AGP are also used as surrogate models for markpainting. Since the baseline methods do not exploit adversarial position in their attack, the locations of watermark embedding are randomly selected for them, while for AWD-AGP, the locations are specified by WPN in the experiments.

The attack results of AWD-AGP and the baselines are given in Table 1. The metrics in Table 1 are calculated between the mask region of the original output and that of the attacked output, where the mask region is a square-shaped area in which the watermark is embedded in the image. The larger difference between the original and attacked output of a model in the mask region suggests that the attack can better disrupt the watermark removal process. Table 1 shows that AWD-AGP achieves superior performance over the baselines on all target models. As expected, DWV shows better performance on blind watermark removers than that on inpainting-based removers, while markpainting demonstrates better performance on inpainting-based removers, since both attacks are designed to target their specific type of watermark remover. Figs. 6 and 7 are some examples of attack results from AWD-AGP and the baselines. Fig. 6 shows that the blind watermark remover, WDModel can still remove the watermarks under DWV's protection, while they fail to do so under the proposed protection. Though AWD-AGP is an untarget attack, it may cause blind watermark removers to fail to detect watermarks in an image. Consequently, the attack results will retain the watermark, as shown in Fig 6. Fig. 7 shows that the inpainting-based remover, FcFnet can still restore images naturally for markpainting attack, but it can only generate unnatural content for AWD-APG. The experimental results presented in Table 1 indicate that AWD-AGP is effective in protecting watermarked images against both inpainting-based and blind watermark removers in a black-box setting.

4.3 Ablation Study

In this section, we evaluate the efficacy of WPN and the attribution-guided perturbation attack scheme individually. To validate the effectiveness of WPN, we first verify that WPN does output location with higher isolated scores S. Therefore, we calculate the isolated score S from four different positions in watermark-free images. These positions include the top left corner, center, random position, and adversarial position predicted by WPN and their S are 0.202, 0.096, 0.101, 0.397, respectively. The results show that the adversarial positions predicted by WPN have the highest isolated score. To evaluate the difficulty level of restoring content in various positions for inpainting models, we compare the restoration results for masks in the four different positions with different inpainting models. As shown in Table 2, the restoration results from the adversarial position generated by WPN are always the worst, indicating the proposed watermark embedding position is more challenging for inpainting-based removers. In addition, we use an ensemble attack to compare the attack performance of the four positions. The objective function and the same set of surrogate models used by AWD-AGP in Table 1 and Table 3 in the appendix are employed to generate adversarial perturbations for a corresponding target model. PGD [22] is utilized to update the perturbations. As shown in Table 2, the adversarial position generated by WPN offers a stronger attack. It demonstrates that the effectiveness of an attack can be enhanced by utilizing the watermark embedding position generated by WPN.

To evaluate the effectiveness of the proposed attribution-guided perturbation scheme (AGP), we apply it to images with watermarks embedded in random positions. Table 1 and Table 3 show that even without the adversarial location, the attribution-guided perturbation scheme still outperforms the baselines. Moreover, we compare the attack performance of AWD-AGP with an attack performed by simply applying PGD on the same set of surrogate models with the watermark embedded in the same position generated by WPN. These PGD attacks are labeled as 'ensemble' in Table 1 and Table 3. As shown in Table 1 and Table 3, AWD-AGP outperforms the simple ensemble attack in almost all metrics except for the L_{ppl} on Matnet. These results demonstrate that the proposed attribution-guided perturbation scheme improves performance over different types of watermark removers in a black-box setting.

5 CONCLUSION

We introduce AWD-AGP, the first watermark protection method capable of defending against both inpainting-based and blind watermark removers simultaneously in a black-box setting. AWD-AGP comprises two primary components: adversarial location for watermark embedding and attribution-guided adversarial perturbation generation. To effectively search for the adversarial location, we propose a novel training scheme that trains a Watermark Positioning Network to predict the optimal location for watermark placement, thereby making watermark removal difficult for inpainting-based removers. Furthermore, since inpainting-based and blind watermark removers exploit information in different regions of an image, we propose an attribution-guided perturbation scheme that identifies those regions and assigns attack strengths to different pixels against different removers. Consequently, the generated perturbation can simultaneously defend against both types of watermark removers. The effectiveness of AWD-AGP has been demonstrated under a black-box setting through experiments on seven models, including four inpainting-based and three blind watermark removers. A comprehensive ablation study confirms the effectiveness of each component of AWD-AGP.

ACKNOWLEDGMENTS

This work is conducted at ROSE @ NTU, Interdisciplinary Graduate Programme, Nanyang Technological University, Singapore, supported by NTU Internal Funding - Accelerating Creativity and Excellence (NTU-ACE2020-03).

REFERENCES

[1] Anurag Arnab, Ondrej Miksik, and Philip HS Torr. 2018. On the robustness of semantic segmentation models to adversarial attacks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 888–897.

[2] Zhiyi Cao, Shaozhang Niu, Jiwei Zhang, and Xinyi Wang. 2019. Generative adversarial networks model for visible watermark removal. *IET Image Processing*, 13, 10, 1783–1789.

[3] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*. Ieee, 39–57.

[4] Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Chau. 2019. Shapeshifter: robust physical adversarial attack on faster r-cnn object detector. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18*. Springer, 52–68.

[5] Xiaodong Cun and Chi-Man Pun. 2021. Split then refine: stacked attention-guided resunets for blind single image visible watermark removal. In *Proceedings of the AAAI Conference on Artificial Intelligence* number 2. Vol. 35, 1184–1192.

[6] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2015. The pascal visual object classes challenge: a retrospective. *International journal of computer vision*, 111, 98–136.

[7] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

[8] Hao Huang, Yongtao Wang, Zhaoyu Chen, Yuheng Li, Zhi Tang, Wei Chu, Jingdong Chen, Weisi Lin, and Kai-Kuang Ma. 2021. Cmua-watermark: a cross-model universal adversarial watermark for combating deepfakes. *arXiv preprint arXiv:2105.10872*.

[9] Yi Huang and Adams Wai-Kin Kong. 2022. Transferable adversarial attack based on integrated gradients. *arXiv preprint arXiv:2205.13152*.

[10] Yi Huang, Adams Wai-Kin Kong, and Kwok-Yan Lam. 2019. Adversarial signboard against object detector. In *BMVC*, 231.

[11] Yi Huang, Fan Wang, Adams Wai-Kin Kong, and Kwok-Yan Lam. 2020. New threats against object detector with non-local block. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*. Springer, 481–497.

[12] Jitesh Jain, Yuqian Zhou, Ning Yu, and Humphrey Shi. 2023. Keys to better image inpainting: structure and texture go hand in hand. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 208–217.

[13] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*. Springer, 694–711.

[14] David Khachaturov, Ilia Shumailov, Yiren Zhao, Nicolas Papernot, and Ross Anderson. 2021. Markpainting: adversarial machine learning meets inpainting. In *International Conference on Machine Learning*. PMLR, 5409–5419.

[15] Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. 2022. Mat: mask-aware transformer for large hole image inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10758–10768.

[16] Xiang Li, Chan Lu, Danni Cheng, Wei-Hong Li, Mei Cao, Bo Liu, Jiechao Ma, and Wei-Shi Zheng. 2019. Towards photo-realistic visible watermark removal with conditional generative adversarial networks. In *Image and Graphics: 10th International Conference, ICIG 2019, Beijing, China, August 23–25, 2019, Proceedings, Part I 10*. Springer, 345–356.

[17] Jing Liang, Li Niu, Fengjun Guo, Teng Long, and Liqing Zhang. 2021. Visible watermark removal via self-calibrated localization and background refinement. In *Proceedings of the 29th ACM International Conference on Multimedia*, 4426–4434.

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 740–755.

[19] Xinwei Liu, Jian Liu, Yang Bai, Jindong Gu, Tao Chen, Xiaojun Jia, and Xiaochun Cao. 2022. Watermark vaccine: adversarial attacks to prevent watermark removal. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIV*. Springer, 1–17.

[20] Yang Liu, Zhen Zhu, and Xiang Bai. 2021. Wdnet: watermark-decomposition network for visible watermark removal. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 3685–3693.

[21] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. 2017. Delving into transferable adversarial examples and black-box attacks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=Sys6GJqxl.

[22] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rJzIBfZAb.

[23] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Z Qureshi, and Mehran Ebrahimi. 2019. Edgeconnect: generative image inpainting with adversarial edge learning. *arXiv preprint arXiv:1901.00212*.

[24] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. 2016. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*.

[25] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

[26] Edgar Riba, Dmytro Mishkin, Daniel Ponsa, Ethan Rublee, and Gary R. Bradski. 2020. Kornia: an open source differentiable computer vision library for pytorch. In *Winter Conference on Applications of Computer Vision*.

[27] Nataniel Ruiz, Sarah Adel Bargal, and Stan Sclaroff. 2020. Disrupting deepfakes: adversarial attacks against conditional image translation networks and facial manipulation systems. In *European Conference on Computer Vision*.

[28] Pu Sun, Yuezun Li, Honggang Qi, and Siwei Lyu. 2020. Landmark breaker: obstructing deepfake by disturbing landmark extraction. In *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*.

[29] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*. PMLR, 3319–3328.

[30] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

[31] Zuxuan Wu, Ser-Nam Lim, Larry S Davis, and Tom Goldstein. 2020. Making an invisibility cloak: real world adversarial attacks on object detectors. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*. Springer, 1–17.

[32] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. 2017. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE international conference on computer vision*, 1369–1378.

[33] Fengting Yang, Qian Sun, Hailin Jin, and Zihan Zhou. 2020. Superpixel segmentation with fully convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13964–13973.

[34] Chin-Yuan Yeh, Hsi-Wen Chen, Hong-Han Shuai, De-Nian Yang, and Ming-Syan Chen. 2021. Attack as the best defense: nullifying image-to-image translation gans via limit-aware adversarial attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

[35] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. 2018. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5505–5514.

[36] Tao Yu, Zongyu Guo, Xin Jin, Shilin Wu, Zhibo Chen, Weiping Li, Zhizheng Zhang, and Sen Liu. 2020. Region normalization for image inpainting. In *Proceedings of the AAAI conference on artificial intelligence* number 07. Vol. 34, 12733–12740.

[37] Yu Zeng, Zhe Lin, Huchuan Lu, and Vishal M Patel. 2021. Cr-fill: generative image inpainting with auxiliary contextual reconstruction. In *Proceedings of the IEEE/CVF international conference on computer vision*, 14164–14173.

A COMBINING SUPERPIXELS TO SUPERPIXEL CULSTERS

Given superpixels $s_1, s_2, \cdots, s_n$, where n is the number of superpixles. First, we compute the mean μ_i and variance σ_i of pixel values of the i^{th} superpixel for $i = 1, 2, \cdots, n$. Next, we normalize μ_i and variance σ_i based on the following equation:

$$\begin{aligned} \mu_i' &= \frac{\mu_i - \mathrm{E}(\boldsymbol{\mu})}{\mathrm{Var}(\boldsymbol{\mu})}, \\ \sigma_i' &= \frac{\sigma_i - \mathrm{E}(\boldsymbol{\sigma})}{\mathrm{Var}(\boldsymbol{\sigma})}, \end{aligned} \tag{9}$$

where $\boldsymbol{\mu} = (\mu_1, \mu_2, \cdots, \mu_n)$ and $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \cdots, \sigma_n)$. $\mathrm{E}(\cdot)$ and $\mathrm{Var}(\cdot)$ represent the expectation and variance of a variable, respectively. Here we combine superpixels based on their statistical

Table 3: The additional attack results of AWD-AGP and the baselines.

Target Model	Surrogate Models	Position	Method	SSIM ↓	PSNR ↓	L_{ppl} ↑	RMSE ↑	REMSE ↑
DBWEModel	Matnet FcFnet	Random	Markpainting	0.862	21.058	0.665	0.106	0.254
	WDModel SLBRModel	Random	DWV	0.834	20.782	0.755	0.111	0.279
	Matnet FcFnet WDModel SLBRModel	Random	AGP	**0.812**	20.783	0.905	0.109	0.295
	Matnet FcFnet WDModel SLBRModel	WPN	Ensemble	0.828	19.589	0.868	**0.120**	0.299
	Matnet FcFnet WDModel SLBRModel	WPN	AWD-AGP	0.815	**19.559**	**0.950**	**0.120**	**0.310**
Gennet	Matnet FcFnet	Random	Markpainting	0.552	16.119	1.634	0.180	0.369
	WDModel SLBRModel	Random	DWV	0.729	22.597	0.921	0.094	0.327
	Matnet FcFnet WDModel SLBRModel	Random	AGP	0.522	16.127	1.713	0.186	0.380
	Matnet FcFnet WDModel SLBRModel	WPN	Ensemble	0.554	16.669	1.639	0.171	0.382
	Matnet FcFnet WDModel SLBRModel	WPN	AWD-AGP	**0.484**	**14.736**	**1.912**	**0.206**	**0.399**
Crfillnet	Matnet FcFnet	Random	Markpainting	0.757	21.792	0.835	0.097	0.315
	WDModel SLBRModel	Random	DWV	0.858	27.018	0.485	0.052	0.283
	Matnet FcFnet WDModel SLBRModel	Random	AGP	0.738	21.614	0.885	0.101	0.324
	Matnet FcFnet WDModel SLBRModel	WPN	Ensemble	0.757	22.471	0.898	0.088	0.343
	Matnet FcFnet WDModel SLBRModel	WPN	AWD-AGP	**0.709**	**20.574**	**1.055**	**0.109**	**0.359**

Algorithm 1: Combining Superpixels

Input: superpixels $s_1, s_2, \cdots, s_n$
Output: superpixels clusters $c_1, c_2, \cdots, c_m$

```
for i = 1 to n do
    compute mean μ_i and variance σ_i of pixel values in s_i
end
for i = 1 to n do
    initialize c_i as a cluster containing s_i ;
    compute normalized μ'_i and σ'_i based on equation (9) ;
    initialize f_{c_i} based on equation (10)
end
k= n ;
while do
    k= current number of superpixel clusters ;
    for i = 1 to k do
        if c_i is not empty then
            for c_j in Neighbhood(c_i) do
                if c_j is not empty and MSE(f_{c_i},f_{c_j}) < 0.1 then
                    put all superpixels of c_j to c_i ;
                    mark c_j as empty cluster ;
                    update f_{c_i}
                end
            end
        end
    end
    remove all empty clusters ;
    if no cluster is merged then
        break
    end
end
```

features. To explain the combining process, we first define the features of a superpixel cluster c_j as below:

$$f_{c_j} = \left(\frac{\sum_{s_i \in c_j} \mu_i' \times a_i}{\sum_{s_i \in c_j} a_i}, \frac{\sum_{s_i \in c_j} \sigma_i' \times a_i}{\sum_{s_i \in c_j} a_i}\right), \tag{10}$$

where a_i is the number of pixels in s_i and f_{c_j} is computed by the weighted average of the statistical features of all the s_i that belong to the c_j. Initially, each superpixel is taken as a superpixel cluster. The mean square error (MSE) between the features of each two adjacent clusters, as defined in equation (10), is calculated. If the MSE between f_{c_j} and f_{c_u} is less than 0.1, the two clusters are merged, where c_u is an adjacent cluster of c_j. We iterate through each pair of adjacent clusters, calculate their MSE, and merge them if the merge condition is satisfied. After each merge, we update the cluster features and repeat the process until the number of clusters no longer changes. The algorithm for combining superpixels is given in Algorithm 1.

B ADDITIONAL INFORMATION AND EXPERIENTIAL RESULTS

The links to download the four pre-trained inpainting watermark removers are listed below:

- Matnet: https://github.com/fenglinglwb/MAT
- FcFnet: https://github.com/SHI-Labs/FcF-Inpainting
- Gennet: https://github.com/JiahuiYu/generative_inpainting
- Crfillnet: https://github.com/zengxianyu/crfill