# Chimera: Creating Digitally Signed Fake Photos by Fooling Image Recapture and Deepfake Detectors

Seongbin Park, Alexander Vilesov, Jinghuai Zhang, Hossein Khalili, Yuan Tian, Achuta Kadambi, and Nader Sehatbakhsh, *University of California, Los Angeles*

https://www.usenix.org/conference/usenixsecurity25/presentation/park

This paper is included in the Proceedings of the
34th USENIX Security Symposium.

August 13–15, 2025 • Seattle, WA, USA

978-1-939133-52-6

# Chimera: Creating Digitally Signed Fake Photos by Fooling Image Recapture and Deepfake Detectors

Seongbin Park*, Alexander Vilesov*, Jinghui Zhang, Hossein Khalili,
Yuan Tian, Achuta Kadambi, Nader Sehatbakhsh
*University of California, Los Angeles*
**{parkseongbin,vilesov}@ucla.edu*

## Abstract

Deepfake detectors relying on heuristics and machine learning are locked in a perpetual struggle against evolving attacks. In contrast, cryptographic solutions provide strong safeguards against deepfakes by creating hardware-binding digital signatures when capturing (real) images. While effective, they falter when attackers misuse cameras to recapture images of digitally generated fake images from a display or other medium. This vulnerability reduces the security assurance back to the effectiveness of deepfake detectors. The main difference, however, is that a successful attack must now deceive two types of detectors *simultaneously*: deepfake detectors and detectors specialized for detecting image recaptures.

This paper introduces *Chimera*, an end-to-end attack strategy that crafts cryptographically signed fake images capable of deceiving both deepfake and image recapture detectors. Chimera demonstrates that current adversarial and generative models fail to effectively deceive both detector types or lack generalization across different setups. Chimera addresses this gap by using a hardware-aware adversarial compensator to craft fake images that successfully bypass state-of-the-art detection mechanisms. The key innovation is a GAN-based image generator that accounts for and compensates the physical transformations introduced during the recapture process. Through rigorous testing using commercial off-the-shelf cameras and displays, Chimera proves effective in fooling both types of detectors with a high success rate while having high visual quality (compared to the original real image). Chimera demonstrates the vulnerability of deepfake detectors even when equipped with hardware-based digital signatures. Our successful end-to-end attack on state-of-the-art detectors shows an urgent need for more robust detection and mitigation strategies. The source code is available at `https://github.com/ssysarch/Chimera`.

## 1 Introduction

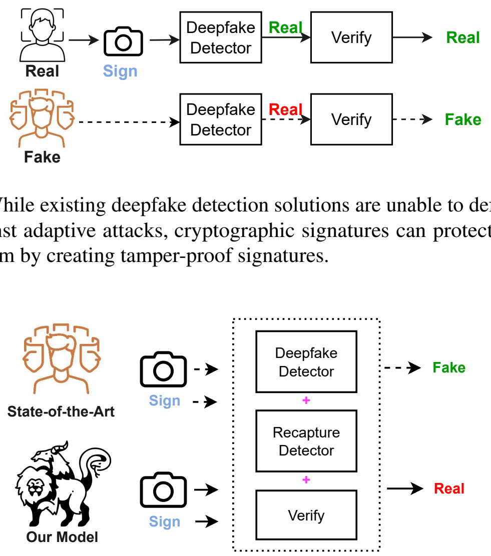
Recent progress in generative machine learning research has significantly improved the quality of synthetic media created by such models. In the image domain, the development of Generative Adversarial Networks (GANs) [19, 27–29] and diffusion models [15, 22, 43] has enabled various real-world applications, such as generating medical or private training data [18, 25]. However, there have also been concerns that these techniques can be used to generate manipulative and abusive content [10, 37]. For example, very recently, a slew of fake images have been used to deceive the public and influence the 2024 US election [20].

The importance of this problem has led to a flurry of deepfake detector proposals [3, 30, 38, 40, 46, 57, 61]. The core idea is to leverage various heuristics, most commonly supervised machine learning, to distinguish fake from real images. However, the fundamental limitation of these solutions is that they struggle to protect the system against evolving attacks.

A more effective solution to combat digitally created fake images is to utilize cryptography and hardware. Specifically, digital signatures are increasingly being used in commercial cameras [1]. The goal is to embed digital signatures, generated by camera hardware, in images so they can be differentiated from digitally fabricated ones. These tamper-resistant digital signatures will include details such as date, time, location, a hashed version of the image content, and even the photographer's information [2]. Such a technique can then effectively split images into two categories: images guaranteed to have been taken by a physical camera and images whose source cannot be verified.

An important observation is that cryptography-based solutions are *vulnerable to physical source manipulations* in which the camera is used to capture a screenshot from a digitally generated fake image (instead of taking a picture from a "real" scene). While a cryptography-based solution can guarantee that images are produced by a camera, the content of the image is not necessarily of a real scene. We define a "fake" image as any digitally generated image displayed on a

*The first two authors contributed equally (the author order does not reflect their extent of contributions).



(a) While existing deepfake detection solutions are unable to defend against adaptive attacks, cryptographic signatures can protect the system by creating tamper-proof signatures.

(b) Cryptographic methods are vulnerable to *screenshot deepfake* attacks. Successful attacks, however, need to fool both deepfake AND recapture detectors. Our scheme, *Chimera*, is the first method to achieve such a capability.

Figure 1: Comparison between existing deepfake attacks and *screenshot deepfake* attacks for state-of-the-art and **our method**. A red label means that the classifier is fooled.

screen and captured by a camera, while "real" images consist of all other images. A more detailed description is provided in Section 3.1. This vulnerability, which we call ***screenshot deepfake***, allows an adversary to generate arbitrary images and digitally sign them by taking a screenshot. We assume that the adversary will typically utilize a deepfake generator as a source for "fake" images to exploit this vulnerability.

There are two potential solutions for this attack: ***(i)*** leveraging *deepfake detectors* as the underlying source is still a fake image and ***(ii)*** using *image recapture detectors*, commonly used to detect image recapture [4, 11, 58]. Crucially, the adversary must fool BOTH types of detectors simultaneously to launch a successful *screenshot deepfake* attack.

In this paper, we present a *screenshot deepfake* attack scheme called *Chimera*. Our attack is capable of generating arbitrary fake images, digitally signed by a genuine camera, that can fool both state-of-the-art deepfake and image recapture detectors when a commercial off-the-shelf camera is used to take the screenshot. The ***key idea*** is developing a two-step attack strategy where the transformation caused by image recapture can be learned and hence compensated for by the model. A brief overview of our approach is shown in Figure 1.

*Chimera* has to overcome several research challenges as existing deepfake generators utilized in this attack scheme are unable to reliably fool both deepfake and image recapture detectors. The main challenge in our design is that an image recapture of a display induces artifacts such as Moiré patterns which can be visible to the human eye or detected by recapture detectors. To address this, we develop an attack that is personalized to a camera and display pair: first, we show that through careful adjustment of camera settings, such as changing the camera's focus, unwanted artifacts of screen recapture can be largely mitigated, and second, we show that by making alterations to how an image is displayed, we can fool recapture detectors and even downstream deepfake detectors. This is achieved by designing a two-way GAN network that first learns the transformation caused by the screen recapturing process and then compensates for artifacts from the process by a generator network.

In short, compared to existing methods, *Chimera* solves the following new challenges:

***(i)*** It develops a method that deceives not only recapture detectors but also downstream detectors of deepfakes. *Chimera* does not try to fool each detector in isolation, instead, it tries to create a digitally signed fake image that can fool both detectors simultaneously.

***(ii)*** It creates an adversarial attack that indirectly fools a detector. This is in contrast with established adversarial machine learning attacks where the perturbation was directly applied to the input of the target classifier. More specifically, in *Chimera*, the perturbation is applied to the fake image, $X$, which creates $\tilde{X}$. The adversarial image, however, is then transformed to $g(\tilde{X})$ due to the recapturing. The transformed image, $g(\tilde{X})$, now needs to fool *both* recapture detectors and deepfake detectors. This is different from fooling one classifier directly (with $\tilde{X}$) in existing methods.

We evaluate our attack strategies on two state-of-the-art deepfake detectors and three image recapture detectors. We leverage different physical setups including different cameras and displays, as well as various configurations for the camera. Results show that *Chimera* can reduce the detection accuracy of state-of-the-art recapture and deepfake detection by more than 50% while increasing the success rate of fooling a layered defense scheme (both deepfake and recapture detector) by about 15%.

In short, the contributions of this paper are:

- We develop a new end-to-end attack strategy that can circumvent state-of-the-art defense mechanisms against deepfakes including cryptography, deepfake detectors, and image recapture detectors.
- We introduce a new technique that is hardware-aware, capable of adapting to unique combinations of a camera and display pair, to address the challenges in *screenshot deepfake* attacks.
- We evaluate our results in real-world settings using various detectors and configurations. Our models and experiments are publicly available.

## 2 Background

### 2.1 Deepfake

**Overview.** Deepfakes are rapidly increasing and causing real societal threats including fake news articles [52], politically-motivated videos [39], fooling facial liveness verification [31], and impacting financial systems [55]. As the creation of deepfakes becomes increasingly simple with the availability of open-source tools, their negative impact on society is becoming more apparent. Consequently, deepfake detection is now an essential responsibility for governments and industries.

**Image Generation.** Recent advancements in machine learning have made deepfake generation increasingly more powerful. Deepfakes are mostly based on generative models such as GANs [27], Variational Autoencoders (VAEs) [54], and Diffusion models [43,45]. Recently, foundation models have significantly enhanced both the quality and user-friendliness of deepfake generators [14,16,41,42]. Lastly, deepfake creation has become more accessible due to the introduction of crowd-sourced websites such as *Huggingface* and *CivitAI*. According to a recent review [3], more than 3,000 user-customized image generation models exist on the Internet.

**Deep-Learning Detection.** As deepfake technology becomes more and more realistic, detecting such synthetic photos has become increasingly critical. Researchers have developed several learning-based methods for deepfake detection [40,57,61]. Lgrad [49] employs a pre-trained CNN model to convert images into gradients, which are leveraged as forgery artifacts that can be detected by a classifier. Ojha *et al.* employs a frozen vision-language model (CLIP-ViT) to extract forgery features [38]. Most recently, Fatformer [35] adapts features within both image and frequency domains and uses contrastive objectives between the adapted features and text prompt embeddings to identify forgery traces.

Despite their high performance on existing deepfake models, current detectors struggle to generalize against emerging threats from user-customized generative models and vision foundation models [3]. Additionally, deepfake detectors and generators are locked in an arms race, making detectors an inherently imperfect, always-evolving system rather than a fully robust solution.

**Image Provenance/Verification.** A more robust solution for detecting deepfake is through verifying image provenance - i.e., certifying the source and history of media content (e.g., image). Verifying an image involves implementing a protocol that cryptographically signs an image at the moment of capture, embedding provenance information that serves as proof of the image's authenticity [12,59]. The signatures are created in the camera, using tamper-resistant hardware and/or a trusted software module. Additional edits to the image could also be securely signed, allowing the end user to reason about the origin and lifetime of an image.

Organizations such as Content Provenance and Authenticity (C2PA) develop technical standards that can enable such a solution in the real world [1,44]. Many commercial camera manufacturing companies (e.g., Canon, Nikon, Sony, etc.) have already built cameras with this capability [2].

While standards such as C2PA may be able to verify that an image truly came from a camera, spoofing the system through image recapture is an emerging threat [56]. Specifically, while the camera and hardware can safeguard the image generation pipeline *after* an image is created, they cannot protect the system from physical alterations to the scene *before* the image is captured. Consequently, an adversary can perform a *screenshot deepfake* attack by placing a fake image in front of a regular camera. The camera, unaware of this malicious manipulation, captures an image and signs it. As a result, the image is signed but contains false data, effectively undermining the usefulness of data provenance and signatures.

### 2.2 Recaptured Image

**Overview.** Image recapture is a common method for cybertheft [13]. Typically, this is conducted by malicious insiders who use their cameras/smartphones to photograph the secret files displayed on screens. Additionally, screen recapture could help the attacker to successfully remove the embedded hidden digital watermarks due to the optical noises introduced during photographing.

As discussed earlier, another crucial concern with image recapture is that it can produce legitimate cryptographic image signatures since the camera has no way to distinguish between a regular image capture and a screen recapture. When used to recapture a deepfake image, this creates a *screenshot deepfake* attack which is the focus of this paper.

**Image Recapture Detection.** Primarily used for forensic analysis, several methods have been proposed to detect recaptured images. Cao *et al.* [11] extracted texture, detail loss, and color features from images to feed into a probabilistic SVM. Thongkamwitoon *et al.* [51] developed an algorithm based on learned edge blurriness and distortion features using K-singular value decomposition. *Yang et al.* [58] introduced a generalized model for small-size recapture image forensics using Laplacian Convolutional Neural Networks, achieving over 95% detection accuracy on all image sizes (up to 99% for larger images). *Li et al.* [32] proposed a highly effective method targeted toward detecting all types of recaptured images; it involves a hierarchical strategy combining CNNs and RNNs to exploit both intra-block information and inter-block dependencies. Agrawal *et al.* [5] compiled a vast and varied dataset of rebroadcast images, demonstrating the robustness of Markov-based methods and the superior performance of a convolutional neural network (CNN) based approach in identifying rebroadcast attacks.

Abraham *et al.* proposed a moiré pattern detection network [4]. This model decomposes images via a Wavelet decomposition and then processes them through a multi-input

CNN. A key strength of this approach is the use of the LL intensity image (from the Wavelet decomposition) as a weight parameter for the moiré pattern, allowing it to distinguish effectively between high-frequency background textures and moiré patterns. Cheng *et al.* utilized Moiré patterns for watermarking digital content in order to detect the source of confidential digital content through camera recapture [13]. More recently, Li *et al.* proposed a novel two-branch deep neural network that leverages multi-scale cross-attention fusion to fuse RGB and frequency information, improving generalization across various recapture scenarios [33]. The first branch extracts detail loss artifacts using a frequency filter bank preprocessing module, and the second branch identifies color distortion artifacts from the RGB input. The final predictions are generated by fusing the discriminative features from both the frequency and RGB modalities.

There are adaptive attacks against these detection methods, but they involve direct manipulation of the image to bypass forensic analysis [17]. Since these manipulations will alter the manifest/signature bound to the image, they do not apply to our purposes.

## 3 Threat Model

### 3.1 Attacker Goal

There are four categories of images one can generate:

- **Raw Real Images:** Authentic images captured directly by a camera without any manipulation or processing. These images are genuine and would be recognized as such once signed with the cryptographic protocol outlined in Section 2.1 if such capability exists in the target camera.

- **Raw Fake Images:** Synthetic images generated using generative machine learning models. These images are fake and (ideally) would be recognized as such due to either the lack of a cryptographic signature or using a fake image detector.

- **Recaptured Real Images:** Genuine images that have been displayed on a screen or another medium and then re-photographed. These images would be signed as real under the protocol, however, they (ideally) would be detected as recaptured using an image recapture detector.

- **Recaptured Fake Images:** Deepfake or synthetic images that have been displayed on a screen and then photographed or scanned to create a new image. Despite being fake, these images would be signed as real under the cryptographic protocol. However, they can be labeled as fake or recaptured (or both) using fake and recapture detectors.

The attacker therefore seeks to deceive the public (e.g., false advertisement, misinformation, etc.) by generating a ***recaptured fake image*** that can bypass detection mechanisms and be cryptographically signed as an authentic, real image. For the attack to be successful, the attacker must ensure that the recaptured image is perceptually indistinguishable from genuine photographs and successfully fools both recaptured image detectors and deepfake detection algorithms.

### 3.2 Target Models

We target several recaptured image detectors, including those proposed in prior works [4, 33], which have been trained on various publicly available datasets [5, 11, 51], as well as on image pairs captured using the exact camera and monitor setup of the attacker.

The first model, **TwoB_DCT**, is the original implementation by Li *et al.* [33], as mentioned in Section 2.2. The second model, **TwoB_DWT**, is similar to **TwoB_DCT**, but instead of using a frequency filter bank, it employs Discrete Wavelet Transform (DWT). In this model, the high-frequency information is extracted from the LH, HL, and HH components of the decomposition. The third model, **MoireDet**, is the moiré pattern detection network proposed by Abraham *et al.*, repurposed to detect recaptured images [4].

Additionally, we target two deepfake image detectors, **FatFormer** [35] and **UnivDetect** [38], to evaluate the effect of our attack on their performance.

The criteria for selecting these models were ***(i)*** being state-of-the-art and widely used, ***(ii)*** having publicly available code. All models and data, including these models and our newly developed attack models, are available online.

### 3.3 Attacker Assumptions

In this paper, the target models are black-box systems to the attacker, meaning that the attacker does not have access to the training data, model parameters, or internal workings of any model. This is the most difficult setting to create a successful attack but highly generalizable. The attacker is free to modify and optimize any generated deepfake, but once the photograph of the deepfake is taken, the attacker cannot further alter the image to enhance its chances of passing as authentic. Therefore, the success of the attack relies on carefully crafting the deepfake and setting up the camera and screen to ensure the image meets all necessary criteria without any post-processing.

We further assume that the camera and its hardware, including the signature generation logic, are trustworthy, hence the attacker cannot forge the signatures nor conduct a replay attack. The attacker, however, has access to an arbitrary camera and a display and can take pictures at will. They can also control the configuration of both including what can be displayed in front of the camera and internal configurations of

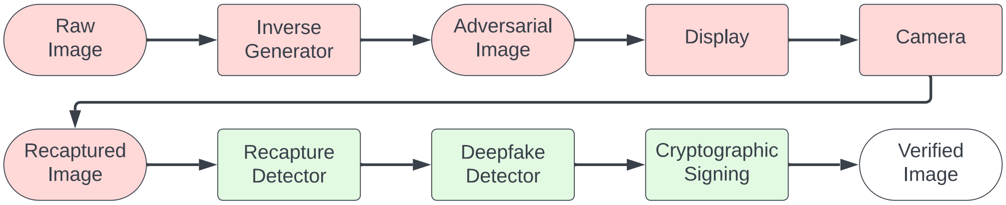


Figure 2: **Block diagram overview of our attack.** We color the components that the attacker and defender have control over. We assume that the attacker has control over the input image, display, and camera parameters, while a defender would implement a recapture and deepfake detector followed by a cryptographic signing.

the camera including changing the focus. The attacker, however, does not have access to the camera's secret key thus the only way to sign an image is by actually taking a picture.

# 4 Design

## 4.1 Overview

We present the details of our attack scheme, *Chimera*, in this section. An overview of its workflow is shown in Figure 2. Overall, the attacker has control over three main parameters: *(i)* the initial fake image, *(ii)* how the image is displayed in front of the camera, *(iii)* how the camera recaptures the image. After this point, the attacker has no other control over any aspect of the attack. As mentioned in Section 3, the attacker does not know the internals and/or architecture of the detectors. More specifically, the attacker first creates a fake image. Standard deepfake image creators could be used here (details in Section 4.2). The attacker then leverages *Chimera inverse generator* to transform the initial fake image into an *adversarial image* (details in Section 4.3.2). They can then control the display and/or camera configurations to generate the final *recaptured image* (see details in Section 4.3.1).

On the defense side, the recaptured image goes through three checks in no particular order: *recapture detection*, *deepfake detection*, and *signature verification*. The attack is successful if and only if it passes all three checks.

## 4.2 Generating a Fake Image

Before an adversary launches the *Chimera* attack, they must first generate a fake image to be recaptured. This can be done with any generative model since our method is independent of the generation process itself. The design and content of the fake image are an orthogonal problem and our method is designed to be generic.

We utilize the dataset released by Wang *et al.* [57], which consists of StyleGAN2 [29] images trained on LSUN [60]. We use three classes of fake and real images from this dataset: cat, church, and horse.

## 4.3 Camera and Display Interactions

**Artifacts in Image due to Display Configuration.** One of the primary artifacts that are induced by a camera capture is the color Moiré. This typically occurs due to sampling mismatch between a camera's pixel structure (Bayer pattern [48]) and a display's pixel structure as well as imperfections in the orientation between a camera and display (e.g., if the camera sensor plane and display are not parallel). Additionally, the behavior of color aliasing is different between red/blue and green pixels due to the difference in spatial sampling frequency occurring from the Bayer pattern [62].

Concretely, one can analyze the spatial sampling frequency of both the display and camera. If the spatial frequency of the display's pixel grid is close to or larger than half of the camera's sampling frequency, the Nyquist frequency, aliasing may occur due to violating Nyquist sampling rate conditions. The Nyquist frequency $f_N$ can be defined by using the distance between camera pixels, $\Delta x$, such that $f_N = \frac{1}{2\Delta x}$. Therefore, if the display's pixel frequency, after projection into camera coordinates, is higher than $f_N$, the camera would be undersampling the image. Accordingly, high-frequency components in the image display may alias to lower frequencies. However, this analysis is further complicated by how color channels interact since different parts of the Bayer pattern may sample different subpixels of the display, which leads to incorrect color interpretation. For example, a uniform color depicted by a display may be recorded due to aliasing as alternating streaks of red, green, and blue, resulting in a visible Moiré pattern.

According to the theory of Moiré phenomena [6], the effect can often be described as the product of two grating functions. In the case of image recapture, the display may be represented by $D_c(x,y)$ and the camera's sampling function by $S_c(x,y)$, where the sampling function may be dependent on a particular color channel, $c$. The resulting image is defined as the product of these two functions $I_c(x,y) = D_c(x,y) \cdot S_c(x,y)$. In the frequency domain, the image captured by the camera is written as the convolution of the display's signal $\mathcal{F}(D(x,y))$ with the camera's sampling function $\mathcal{F}(S_c(x,y))$, leading to:

$$\mathcal{F}(I_{c,Moire}(x,y)) = \mathcal{F}(D_c(x,y)) * \mathcal{F}(S_c(x,y)), \qquad (1)$$

where $\mathcal{F}(\cdot)$ denotes the Fourier transform operator. This convolution introduces aliasing artifacts into the color channels that recapture detectors can utilize for detection.

Without any additional compensation, a recapture detector, $C_{recap} : I \rightarrow \{true, false\}$, can accurately discriminate between a genuine image, $I(x,y)$, and a recaptured image, $I_{c,Moire}(x,y)$ – i.e., $C_{recap}(I(x,y)) = false$ and $C(I_{c,Moire}(x,y)) = true$ due to their fundamentally different representations.

The goal in *Chimera* is to eliminate this unwanted transition by creating a *compensation* function, $f_{Chimera} = I_{recap}(x,y)$, such that for a fake image $I_f(x,y)$, we have: $I_{c,Moire}(x,y) \neq I_{recap}(x,y) \approx I_f(x,y)$. Although not directly tuned for a specific classifier, when applied to (any) $C_{recap}$, the ultimate goal is to achieve $C(I_{recap(x,y)}) = false$.

Furthermore, assuming that $I_f(x,y)$ is originally fooling a deepfake detector, $C_{deep}(i) = \{real, fake\}$, then $I_{recap}(x,y) \approx I_f(x,y)$ indirectly implies $C_{deep}(I_f(x,y)) = C_{deep}(I_{recap}(x,y)) = fake$.

We propose two main steps for designing $f_{Chimera}$. Specifically, we first develop a preliminary *camera-side* modification scheme that applies a *filtering* function to $I_{c,Moire}(x,y)$, creating a new function, $I_{c,Moire,filter}(x,y)$, that is similar to the application of an optical low-pass filter for a camera. Second, our primary contribution is to design a GAN network that transforms the fake image, $I_f(x,y)$, into an adversarial image, $g(I_f(x,y)) = I_{adv}(x,y)$. Collectively, for a given (real or fake) image, $I(x,y)$, we have $f_{Chimera} = I_{recap}(x,y)$:

$$I_{recap}(x,y) = I_{c,Moire,filter}(g(I(x,y))). \quad (2)$$

We describe the details of our model, $f_{Chimera}$, as follows.

### 4.3.1 Camera-side Modifications

A camera's lens system determines the sharpness of the captured image. This sharpness allows more artifacts from the display to enter the captured images, which can be utilized by image recapture detectors. However, a camera's lens system can still be altered to reduce this sharpness by altering the point spread function (PSF) response to a display. The PSF determines which spatial frequencies are transferred through the optics of the camera. Typically, the lens is adjusted to ensure a sharp image with high spatial frequencies. As mentioned in Section 4.3, when capturing a display, this can harm image quality due to introducing visible artifacts. However, when capturing an image of a display, we can reduce these artifacts through blurring by ***deliberately altering the focus*** to change the spatial frequency content of the image. A similar technique was applied in [51] where aliasing was reduced due to diffraction by reducing the diameter of the aperture instead of the focusing range.

This is an important aspect of *Chimera*, where we change the focusing range of the lens to effectively create a natural low pass filter. This blur can be mathematically represented by convolving the original image with a point spread function (PSF), $H(x,y)$, such that the resulting image is defined by $I_c(x,y) = (D_c(x,y) * H(x,y)) \cdot S_c(x,y)$. In the frequency domain, this is equivalent to multiplying the Fourier transform of the image by the Fourier transform of the PSF leading to:

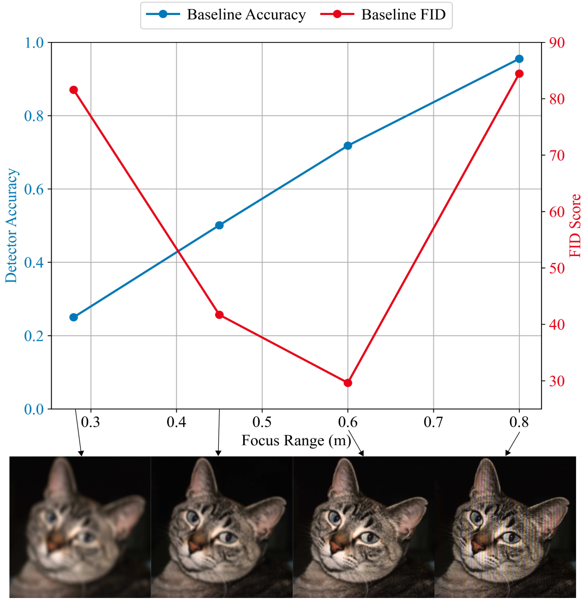


Figure 3: **Camera-side modifications such as adjusting camera capture parameters affect recapture detection accuracy.** We show that adjusting the focusing range of the camera changes the detector accuracy (MoireDet [4]) and perceptual quality (FID score). Notice that at a focus range of $0.8m$, the display is in perfect focus, but yields Moire patterns and a poor FID score. Deliberately lowering the focus range lowers detector accuracy, however, too much blur due to out of focus also leads to poor perceptual quality.

$$\begin{aligned}\mathcal{F}(I_{c,Moire,filter}(x,y)) = (\mathcal{F}(D_c(x,y)) \cdot \mathcal{F}(H(x,y))) \\ * \mathcal{F}(S_c(x,y)). \quad (3)\end{aligned}$$

Thus, by changing the focusing range appropriately, we can remove frequencies that would alias upon image recapture of the display. Utilizing the focus of a camera can thereby contribute to fooling models that rely on Fourier-based analysis to distinguish between real and display-captured images. Blurring removes information that would help a model determine that an image came from a display. However, there exists a trade-off between using the focus to fool the detectors and creating sharp images. More misalignment in the focus parameter will lead to lower accuracy for the detector, at the cost of lower-quality blurry images.

We show this effect empirically in Figure 3 for recaptured images without any additional manipulations labeled as *baseline* tested with a baseline detector, MoireDet [4]. We can see that there exists an optimal focusing range point where perceptual quality is high yet still lowers the detector's accuracy. However, increasing blur beyond this will lower detector

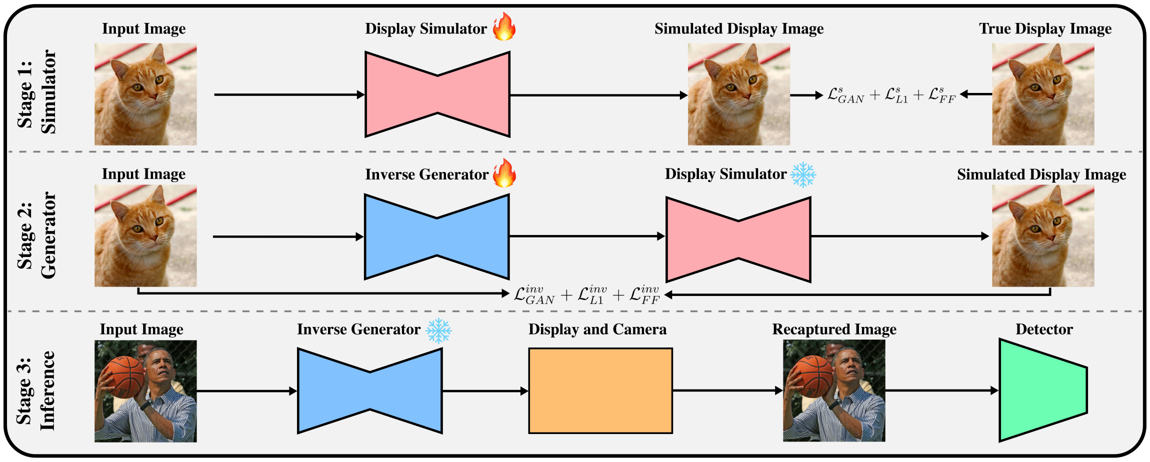


Figure 4: **Our display-side modifications consist of the three stages.** In the first stage we train a *simulator* to learn how a camera would display a particular image. Then, it trains an inverse generator to learn how an input image should be modified so that when displayed it matches the input image as closely as possible. The final stage showcases how *Chimera* is used for attacks.

accuracy at the cost of poorer perceptual quality images.

In the next section, we show that we can create a beneficial trade-off by manipulating the way the image is displayed such that we can still remove most visual artifacts from display capture while maintaining the visual quality of images.

### 4.3.2 Display-side Modifications

The second technique used in *Chimera* involves altering the fake raw image before it is displayed on the screen. This modification increases the likelihood that the recaptured fake image will be free of any artifacts introduced during the recapture process. As a result, the recaptured fake image can effectively deceive recapture detectors.

Let us formally define the recapture process with a fixed camera and screen as $g$. We aim to find an inverse function $g^{-1}$ of the recapture process $g$ such that for any raw fake image $I$, we can have $g(g^{-1}(I)) = I$.

Specifically, we tackle the problem in a two-stage manner. In the first stage, we propose to utilize a neural network to parameterize the image recapture process (i.e., $g$). Then, in the second stage, we learn the inverse function $g^{-1}$ by training another neural network that intends to mitigate the effects of image recapture. We adopt the Conditional Generative Adversarial Network (CGAN) [23] to learn such mappings given the ability of GAN to model any complicated function.

We utilize the CGAN framework to train the two neural networks, one used to simulate the recapture process (called the **Simulator**) and the other used to generate the display-side modifications (called the **Inverse Generator**) with the trained simulator in its loop. The overall framework is shown in Figure 4. In the following, we will explain the Simulator and Inverse Generator processes in detail.

**Simulator.** The simulator $G_s$ is a conditional generative model that takes a raw fake image $I$ and a random parameter $z$ as inputs to generate the recaptured version of $I$. The original implementation of CGAN introduces stochasticity through the use of dropout [47], rather than an explicit random noise parameter. We adopt this approach in our work, and for brevity, will omit $z$ in the remaining sections.

The CGAN framework is used to jointly train the simulator $G_s$ and a discriminator $D_s$ to approximate the image recapture process $g$. Following standard practices of GANs, the discriminator is trained to differentiate between images recaptured by a real camera and those generated by our simulator, conditioned on the corresponding raw images. This process helps to enhance the quality and realism of the simulator. The objective function of learning the simulator $G_s$ with the CGAN framework can be formulated as follows:

$$G_s^* = \arg\min_{G_s}\max_{D_s} \mathcal{L}^s_{GAN}(G_s, D_s) + \lambda_1 \mathcal{L}^s_{L1}(G_s) + \lambda_2 \mathcal{L}^s_{FF}(G_s). \tag{4}$$

We utilize GAN loss, L1 loss, and a focal frequency loss [24] to guide the training process. $\lambda_1$ and $\lambda_2$ represent the weights used to balance different loss terms. In particular, the GAN loss $\mathcal{L}^s_{GAN}$ is used to train the simulator $G_s$ and the discriminator $D_s$ in an adversarial manner, which is expressed as:

$$\begin{aligned}\mathcal{L}^s_{GAN}(G_s, D_s) = &\mathbb{E}_{(I,I_r)}[\log D_s(I, I_r)] + \\ &\mathbb{E}_I[\log(1 - D_s(I, G_s(I))],\end{aligned}$$

where $I$ and $I_r$ represent a raw image and its corresponding ground truth recaptured image. During each iteration, we optimize $G_s$ to minimize this objective against an adversarial $D_s$ that tries to maximize it. Following the literature [23], we also utilize an additional L1 loss $\mathcal{L}_1^s$ to ensure that the simulated image $G_s(I)$ closely resembles the recaptured fake image $I_r$ by a real camera:

$$\mathcal{L}_{L1}^s(G_s) = \mathbb{E}_{(I,I_r)}\left[\|I_r - G_s(I)\|_1\right],$$

Due to the artifacts (e.g., Moiré patterns) introduced during the recapture process, a recaptured fake image exhibits distinctive characteristics in the frequency domain compared to those raw fake images or raw real images. Existing methods for recaptured image detection [4, 7] often rely on frequency domain analysis to identify the recaptured images. Building on this insight, we incorporate the focal frequency loss [24] into our training objective. This loss helps to guide the simulator in more accurately replicating the frequency components commonly found in recaptured images, thereby enhancing the realism of the simulation. Assume the simulator $G_s$ takes a raw fake image with size $H \times W$ as input. Then, the focal frequency loss $\mathcal{L}_{FF}^s$ is expressed as:

$$\mathcal{L}_{FF}^s(G_s) = \mathbb{E}_{(I,I_r)}\left[\sum_{u,v} w(u,v)\left|\mathcal{F}_{I_r}(u,v) - \mathcal{F}_{G_s(I)}(u,v)\right|^2\right],$$

where $w(u,v)$ is the weight for the spatial frequency at spectrum position $(u,v)$. The summation is over all spatial frequencies $(u,v)$ within the image dimensions $H \times W$. We dynamically determine the weights following the same strategy as Jiang *et al.* [24]. Moreover, the spatial frequency of the image $I$ at $(u,v)$ is written as:

$$\mathcal{F}_I(u,v) = \sum_{x=0}^{H-1}\sum_{y=0}^{W-1} I(x,y) \cdot e^{-i2\pi(\frac{ux}{H} + \frac{vy}{W})}.$$

We utilize the UNet architecture as the backbone for our simulator, which adds skip connections between mirrored layers in the encoder and decoder. Moreover, to capture the nuances (e.g., Moire patterns) in the local regions, the Markovian discriminator [23] is employed, which classifies each $N \times N$ patch in an image as real or fake. The final discriminator score is obtained by averaging the results across all the patches. The entire network is trained within the CGAN framework, following the Equation 4.

**Inverse Generator.** The inverse generator $G_{inv}$ is another conditional generative model trained to approximate the inverse function $g^{-1}$ of the recapture process $g$. Specifically, $G_{inv}$ takes in a raw image $I$ to generate an adversarial image $\tilde{I}$, where $G_s(\tilde{I})$ approximates $I$. Once trained, the inverse generator $G_{inv}$ can be used to modify any raw image to generate an adversarial image, allowing the corresponding recaptured image to evade detection. Let $\tilde{I}_r$ denote the recaptured version of the adversarial image $\tilde{I}$. Leveraging the trained simulator $G_s^*$, we can simulate the whole process of obtaining the recaptured fake image $\tilde{I}_r$ as

$$\tilde{I}_r \approx G_s^*(\tilde{I}) = G_s^*(G_{inv}(I)).$$

To this end, we aim to train the inverse generator such that the final simulated image $G_s^*(G_{inv}(I))$ preserves the content of $I$ and does not contain additional patterns that are distinctive to $I$. In this way, the recaptured fake image $\tilde{I}_r$ taken by a real camera will no longer exhibit the artifacts introduced during the recapture process. Similarly, we learn the inverse generator $G_{inv}$ with the CGAN framework, which can be formulated as follows:

$$\begin{aligned} G_{inv}^* = \arg\min_{G_{inv}}\max_{D_{inv}} & \mathcal{L}_{GAN}^{inv}(G_{inv}, D_{inv}) + \\ & \lambda_3\mathcal{L}_{L1}^{inv}(G_{inv}) + \lambda_4\mathcal{L}_{FF}^{inv}(G_{inv}), \end{aligned} \tag{5}$$

where $\mathcal{L}_{GAN}^{inv}$, $\mathcal{L}_{L1}^{inv}$ and $\mathcal{L}_{FF}^{inv}$ represent the GAN loss, L1 loss, and focal frequency loss used to train the inverse generator. $\lambda_3$ and $\lambda_4$ are the weights to balance these loss terms.

The discriminator $D_{inv}$ is trained as a classifier that tries to differentiate between $G_s^*(G_{inv}(I))$ and $I$. Unlike the discriminator $D_s$ used to train the simulator, it operates unconditionally. Specifically, it does not accept $I$ as a conditioning input and instead depends on just the generated $G_s^*(G_{inv}(I))$ (positive samples) or the ground truths $I$ (negative samples). Additionally, we integrate the recapture simulator $G_s^*$ into the training loop as a white-box function and keep it frozen during the training of $G_{inv}$. The detailed formulas of the losses used to train the inverse generator are illustrated as follows. The GAN loss of the inverse generator is expressed as:

$$\begin{aligned} \mathcal{L}_{GAN}^{inv}(G_{inv}, D_{inv}) = & \mathbb{E}_I[\log D_{inv}(I)] + \\ & \mathbb{E}_I[\log(1 - D_{inv}(G_s^*(G_{inv}(I))))]. \end{aligned}$$

The L1 loss of the inverse generator is expressed as:

$$\mathcal{L}_{L1}^I(G_{inv}) = \mathbb{E}_I\left[\|I - G_s^*(G_{inv}(I))\|_1\right].$$

The focal frequency loss of the inverse generator is expressed as:

$$\mathcal{L}_{FF}^s(G_{inv}) = \mathbb{E}_I\left[\sum_{u,v} w(u,v)|\mathcal{F}_I(u,v) - \mathcal{F}_{G_s^*(G_{inv}(I))}(u,v)|^2\right].$$

**Note:** We emphasize that while Chimera is primarily designed to make raw fake images undetectable by detectors, the proposed framework can also be applied to raw real images on the screen (e.g., make raw real images digitally signed by another protocol without taking a photo of the same scene). In other words, Chimera can arbitrarily take in a raw (real or fake) image $I$ to generate an adversarial (real or fake) image $\tilde{I}$ whose recaptured version $\tilde{I}_r$ can fool recapture detectors.

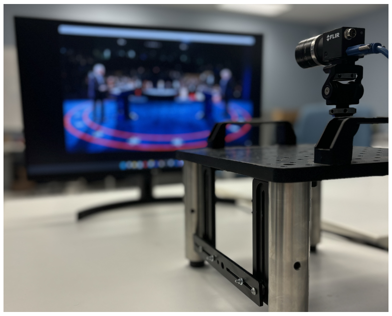


Figure 5: A visualization of one of our **experimental setups** where a camera is focused on a display.

# 5 Evaluation Setup

## 5.1 Metrics

The success of the attack is evaluated based on three key metrics: perceptual quality, performance on deepfake detectors, and performance on recapture detectors.

***Perceptual Quality.*** To assess the visual fidelity of the recaptured samples, we utilize the Fréchet Inception Distance (FID) metric [21]. FID is calculated by comparing the distribution of features between two datasets; in our analysis, both the recaptured samples and the adversarial recaptured samples are compared against the raw samples.

***Recapture Detector Performance.*** The performance of recapture detectors is measured by their accuracy across four distinct image categories outlined in Section 3. This allows us to evaluate how well the detectors can identify recaptured content across real and deepfake images.

***Deepfake Detector Performance.*** The performance of deepfake detectors is likewise assessed on raw images, recaptured images, and recaptured adversarial images. This enables us to assess the detectors' ability to recognize fake content under various conditions.

## 5.2 Experimental Setup

**Datasets.** To evaluate both recapture and deepfake detectors on a single dataset, we chose to utilize a subset of the evaluation set of deepfake detectors [35, 38, 57], specifically the StyleGAN2 [26] images. The dataset consists of four classes: horse, church, cat, and car, which is similar to previous works compared in this paper. For simplicity, the car category was excluded because the images had diverse shapes and were incompatible with our general pipeline. This dataset was split into two parts: one used to train the GANs (as described in Section 4.3.2) and the other used to evaluate recapture and deepfake detectors. The recapture data was obtained by photographing all the images in the dataset (details are provided later in ***Hardware Setup***). For training the recapture detectors, we used a subset of the training dataset from the aforementioned studies [35, 38, 57]. Similarly, recapture data was collected by photographing all the images in this dataset. Lastly, to further evaluate the generalizability of *Chimera* on other datasets, we present the results on 2000 images from a synthetic face dataset [53] generated by StyleGAN [27].

We used separate training and evaluation sets for all experiments. Although the recaptured images of both sets were obtained from the same monitor and camera setup, they contained entirely different raw images. The raw data in our evaluation set were equally split between real and fake images, ensuring balanced representation. To generate recaptured data, we recaptured the same equally split dataset, ensuring that the number of real and recaptured images was also identical.

**Recapture Detectors.** For the evaluation of recapture detection, we employ three distinct model architectures, as described in Section 3.2. When deploying recapture detectors in practice, two main strategies can be considered: training a general model applicable to various cameras and conditions or fine-tuning a base model specifically for each camera. In this paper, we explore both approaches.

***Base Model Fine-Tuning.*** To customize the detector for a particular camera, we begin by training a base model on a general dataset that includes raw and recaptured images from [5, 11, 51]. This base model is then fine-tuned with data from each of the attacker's cameras. This approach avoids the need to retrain a model from scratch each time when adding support for a new camera.

***General Model.*** Alternatively, a generalized model is trained using all the data, combining the general dataset with the data from the attacker's cameras. This approach aims to develop a model that is more robust and performs consistently well across different conditions and devices.

***Blur Augmentations.*** During initial experimentation, we identified that the recaptured image detectors had a strong propensity to classify all blurry or out-of-focus images as recaptured, irrespective of whether the images were raw or genuinely recaptured. This indicates that the models are overreliant on the presence of blur as a key feature indicative of recapture, which in turn leads to a significant increase in false positive rates for raw images that are merely out of focus. Additionally, models trained without blur augmentations consistently classified all recaptured images as such, regardless of focus level. This is unexpected, as substantial blurring typically removes the features distinguishing raw and recaptured images from each other. To mitigate these issues, we introduced blur augmentations into the training pipeline. This modification aims to enhance the model's robustness by reducing its reliance on blur as a distinguishing feature, improving its ability to accurately classify both raw and recaptured images across a spectrum of conditions.

**Deepfake Detectors.** For the deepfake detection component of the evaluation, we use pre-trained models as provided

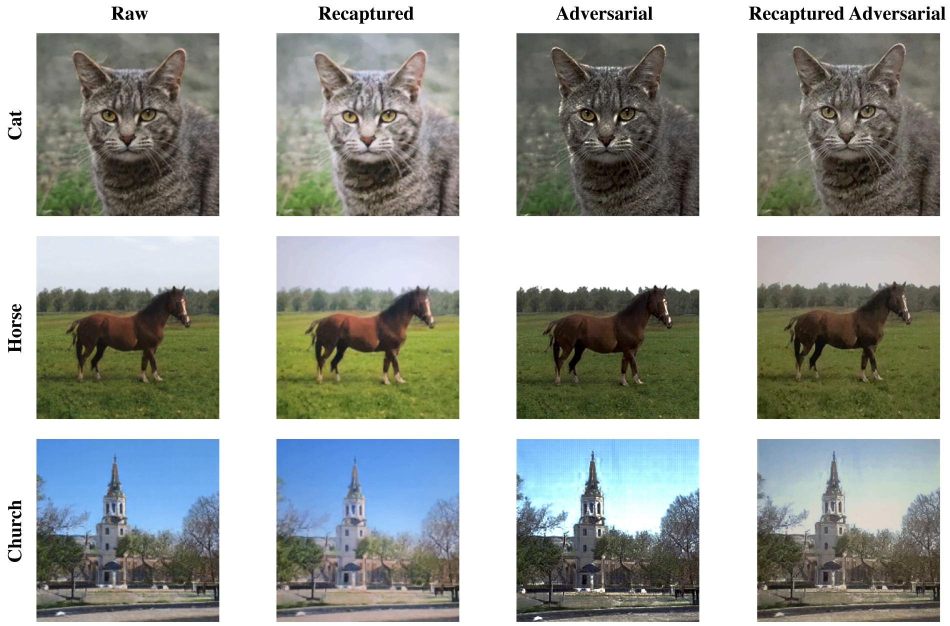


Figure 6: **We showcase the qualitative results of our method.** *Raw* denotes the target image for recapture, *Recaptured* is a baseline recapture image, *Adversarial* is the output image of our inverse generator which when displayed and recaptured appears as the *Recaptured Adversarial* image. Notice how our recaptured adversarial image visually appears closer to the desired target raw image such as brightness and colors. We also include a failure case in the third row where the perceptual quality of our method is poor.

in the literature, leveraging the checkpoints supplied by the respective papers [35, 38]. Each detector is evaluated with a fixed decision threshold of 0.5, allowing for consistent comparison across different detection scenarios.

**Simulator and Inverse Generator** Our implementation modifies the Pytorch implementation of Pix2Pix [23, 64] to do paired uni-directional image translation. In this setup, both the simulator and inverse generator are constructed using a U-Net architecture with 7 downsampling layers, operating on images resized to 1024 pixels.

**Hardware Setup.** While our attack can be implemented on many hardware platforms, we describe for reproducibility the configuration that we used. We test our methods on two different setups. To demonstrate the ease in which our setup can be deployed, we utilize the camera of an iPhone 12 (main camera out of the multi-lens system, 12 MP) and the display of a MacBook Pro (2560×1664 resolution with a density of 224 pixels per inch). The iPhone camera parameters are automatically set by the camera itself during recapture. The majority of results were obtained from this setup unless otherwise mentioned (e.g., Table 3).

We also obtain a second set of results with a highly configurable machine vision camera in order to demonstrate the utility of the camera's focusing range. We use an RGB FLIR Blackfly BFS-U3-51S5P-C camera (1.25 MP) with a manually adjustable focus paired with an LG full HD 31.5 inch display (1920×1080 resolution with a density of 70 pixels per inch). The Blackfly camera's parameters were custom-set to exposure = $20,000$ *ms*, gain = 1, gamma = 0.8, and balance ratio = 1.96. Demosaicing was done manually with bilinear interpolation. All results with this setup are shown in Figure 3, Table 3, and the setup itself is visualized in Figure 5.

To collect large volumes of data efficiently, we synchronized the display of an image with an automatic camera capture with the OpenCV library. After capture, we perform no preprocessing other than cropping to only show the picture on the screen. This is realistic, as users may adjust the window size on many modern phones and cameras before taking a picture. During the training of the simulator and inverse generator, the orientation of the display with respect to the camera is fixed since the simulator is personalized not only to the current camera and display used but also to how they are placed with respect to each other.

## 6 Results

Our results are presented in three categories: Attack success rate against *(i)* Recapture detectors; *(ii)* Deepfake detectors; *(iii)* Layered defenses with both recapture and Deepfake detectors, which is the ultimate goal of *Chimera*.

We evaluate *Chimera* with state-of-the-art deepfake and

| | Raw Real | Raw Fake | Recap Real | Recap Fake | Adv. Recap Real (ours) | Adv. Recap Fake (ours) |
|---|---|---|---|---|---|---|
| Twob_DCT [33] | 0.827 | 0.903 | 0.968 | 0.963 | **0.665** | **0.472** |
| Twob_DWT [33] | 0.867 | 0.930 | 0.942 | 0.952 | **0.418** | **0.283** |
| MoireDet [4] | 0.922 | 0.993 | 0.845 | 0.810 | **0.168** | **0.045** |

Table 1: **Performance of recaptured image detectors** on raw, recaptured, and recaptured adversarial images using a MacBook screen and iPhone. All models exhibit high accuracy for raw and recaptured images. However, we observe a significant drop in accuracy for recaptured adversarial images, indicating that the models are susceptible to our attack (lower is better for attacks).

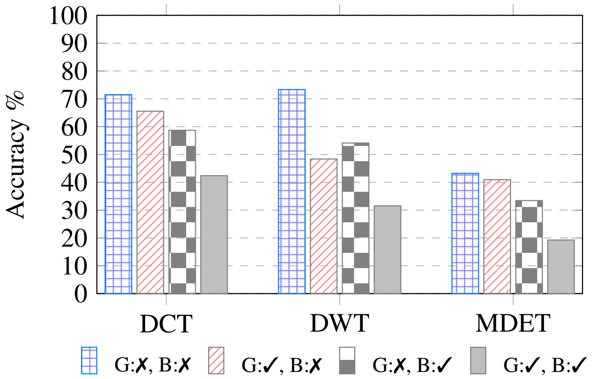


Figure 7: The **impact of using generalization (G) and blurring (B)** on the accuracy of recapture detectors. Lower accuracy means a more successful attack.

recapture detectors. Specifically, we use three recapture detectors (*TwoB_DCT* [33], *TwoB_DWT*, and *MoireDet* [4]) and two deepfake detectors (*FatFormer* [35] and *UnivDetect* [38]) as described in Section 3.2.

For each experiment, we consider three groups of images: raw images, captured images, and images produced by *Chimera* (which we call adversarial recap or adv. recap). Moreover, each group can be generated from real or fake images. There are six cases: raw real, raw fake, recap real, recap fake, adv. recap real, and adv. recap fake.

## 6.1 Qualitative Results

Results for three different classes (cat, horse, and church) are shown in Figure 6. As illustrated in the figures, the perceptual quality of images generated by *Chimera* (last column) remains high compared to raw images. Next, we will discuss the detection accuracy of deepfake and recapture detectors in relation to *Chimera*-generated images. Additionally, we will examine the image quality of *Chimera* and the potential unwanted artifacts it may produce in Section 7.1.

## 6.2 Recapture Detection

The main objective of this section is to study whether images produced by *Chimera* are detectable by state-of-the-art recapture detectors.

**Detection Accuracy of State-of-the-Art Detectors against *Chimera*.** We evaluate the effectiveness of our attack in a realistic scenario using an iPhone camera and two different screens, one in the training data, and one without. Table 1 outlines the different detectors' baseline performance and shows our attack scheme's impact on the detection accuracy of real and fake recaptured images.

Results indicate that raw and recaptured images (both real and fake) are classified with nearly perfect accuracy (close to 100%) by all three detectors. However, the accuracy for images created by *Chimera*, as shown in the last two columns of Table 1, significantly drops. This suggests that the state-of-the-art classifiers struggle to correctly identify these images as recaptured.

Another important observation is that recapture detectors are more effective at identifying fake images than real ones because the fake images were not included in the training set. The detectors are trained on real images, which makes them better at detecting actual recaptures. Since the difference in detection accuracy between the two groups was not significant, we did not retrain the classifiers using fake images.

**Different Displays with Generalization and Blurring.** We evaluate the generalizability of our attack by repeating the above experiment using a different display (LG Monitor) and a camera (Blackfly). The details of our two setups are provided in Section 5.2.

Results are shown in Figure 7. The figure indicates that not using generalization or blurring leads to the lowest attack success rate, which translates to higher detection accuracy for *Chimera*. In contrast, employing both generalization and blurring results in a higher success rate. The key takeaway from this experiment is that *Chimera* is adaptable to different settings, such as various cameras and displays, when trained on more diverse datasets, specifically through blurring augmentation and dataset generalization. Detailed breakdown results are presented in Table 6 and Table 7 in Appendix as part of our ablation study.

Additionally, we report the FID scores for the recaptured and recaptured adversarial images for both screens in Table 2. While the FID scores are higher for the adversarial recaptures, the perceptual quality remains high, as illustrated in Figure 6.

Recall that Figure 3 (cf. §4) illustrated the importance of correctly setting camera parameters and its impact on detector recapture accuracy and perceptual quality measured by FID scores. As depicted in the samples in Figure 3, the

|  | $\text{Recap}_1$ | Adv. $\text{Recap}_1$ | $\text{Recap}_2$ | Adv. $\text{Recap}_2$ |
|---|---|---|---|---|
| FID | 23.439 | 35.827 | 23.699 | 34.094 |

Table 2: **FID scores for images recaptured** on a MacBook screen ($\text{Recap}_1$) and a Monitor screen ($\text{Recap}_2$), along with their corresponding adversarially recaptured images (Adv. $\text{Recap}_1$, Adv. $\text{Recap}_2$).

| Focus Range (m) | 0.28 | 0.45 | 0.60 | 0.80 |
|---|---|---|---|---|
| Baseline Accuracy(%) | 25.0 | 50.1 | 71.8 | 95.5 |
| Baseline FID | 81.58 | 41.69 | 29.63 | 84.44 |
| Attacker Accuracy (%) | 3.2 | 21.5 | 46.5 | 73.6 |
| Attacker FID | 69.87 | 33.93 | 22.8 | 97.21 |

Table 3: Our method **outperforms** the baseline recapture attack in lowering accuracy and improving perceptual quality across most camera focusing ranges (blurs) when evaluated with the Blackfly camera.

Moiré pattern diminishes when defocusing the camera due to the spreading of the PSF. However, after some point, this inevitably causes blurriness and a drop in perceptual quality. We further show the utility of our display-side modifications by training a simulator and inverse generator for each focus level, and we can see a drop in recapture accuracy of approximately 20% over baseline across all focus ranges in Table 3. Lastly, our method also improves the perceptual quality of recaptured images across most focus range settings. Note that on the iPhone, it is not possible to manually adjust focus in fine increments. Instead, we adjusted the camera setup to minimize Moiré patterns without losing too much detail, which can be observed with the low FID scores in Table 2

## 6.3 Deepfake Detection

Results for deepfake detectors are shown in Table 4. Similar to recapture detection experiments, we report the results for raw, recapture, and *Chimera*-generated images for two different setups. In all experiments, we use both blurring and generalization. Accuracy is reported for both real and fake detection (i.e., whether the classifier correctly labeled the image as fake or real) and the overall average.

We consider two state-of-the-art deepfake detectors: UnivDetect [38] and FatFormer [35]. We also enhance the robustness of the deepfake detector by fine-tuning it on recaptured images. Specifically, we focus on fine-tuning UnivDetect with the recaptured images from a MacBook screen. The fine-tuning offers an improvement on the recaptured images, with the fake image detection rate increasing by around 6% for the baseline and adversarial case on both screens. Note that FatFormer does not open-source its training code, so we couldn't further fine-tune it.

| Detector | Dataset | Real | Fake | Average |
|---|---|---|---|---|
| UnivDetect [38] | Raw | 99.83 | 42.00 | 70.92 |
|  | Recap 1 | 98.83 | 27.67 | 63.25 |
|  | Adv 1 | 95.50 | 37.00 | 66.25 |
|  | Recap 2 | 99.67 | 19.00 | 59.33 |
|  | Adv 2 | 96.83 | 20.17 | **58.5** |
| FatFormer [35] | Raw | 100.0 | 96.33 | 98.17 |
|  | Recap 1 | 75.67 | 72.33 | 74.00 |
|  | Adv 1 | 73.83 | 66.33 | 70.08 |
|  | Recap 2 | 88.67 | 66.17 | 77.42 |
|  | Adv 2 | 91.5 | 46.5 | **69.0** |
| Finetuned UnivDetect [38] | Raw | 99.83 | 43.33 | 71.58 |
|  | Recap 1 | 98.83 | 33.33 | 66.08 |
|  | Adv 1 | 94.00 | 44.33 | 69.17 |
|  | Recap 2 | 99.17 | 24.83 | 62.00 |
|  | Adv 2 | 95.5 | 25.00 | **60.25** |

Table 4: **The performance of deepfake detectors** on raw images, recaptured images, and adversarial recaptured images from Screen 1 and 2 (i.e., MacBook and Monitor). Lower accuracy means a more successful attack.

Our results indicate that *Chimera* improves the attack success rate in all scenarios. The success rate (decrease in detection accuracy) ranges from 12% for UnivDetect to approximately 30% in FatFormer.

Another important observation is that while these detectors perform well on raw images, their effectiveness would decrease on recaptured images. For instance, the state-of-the-art deepfake detector FatFormer exhibits a large performance decline, with its average accuracy decreasing by over 20% in such scenarios. Previous research has shown that periodic patterns introduced by recapturing images through a digital screen can adversely affect the performance of deepfake detectors [50]. Our results corroborate these findings, as presented in Table 4. However, as results show, *Chimera* slightly outperforms the recapturing results, showing its effectiveness.

## 6.4 Generalizability of *Chimera*

To further confirm the generalizability of *Chimera* across different datasets, we present the results on 2000 images from a fake face dataset [53] generated by StyleGAN [27]. We use the same setup (MacBook screen and iPhone camera) used in previous sections; note that the GANs trained on the horse, cat, and church classes were not retrained for faces.

Results are presented in Table 5. As can be seen in the table, we observe a significant drop in accuracy when applying *Chimera* to various recapture detectors. The main takeaway from this result is that because our method does not rely on any semantics of the image and instead aims to revert non-semantic artifacts such as color distribution and edge patterns, it is generalizable to other datasets.

|  | Raw | Recap | Adv. Recap |
|---|---|---|---|
| DCT (no Blur) | 0.881 | 0.934 | **0.778** |
| DCT (Blur) | 0.883 | 0.613 | **0.372** |
| DWT (no Blur) | 0.974 | 0.916 | **0.792** |
| DWT (Blur) | 0.908 | 0.746 | **0.574** |
| MoireDet (no Blur) | 0.999 | 0.409 | **0.066** |
| MoireDet (Blur) | 0.952 | 0.671 | **0.140** |

Table 5: **Performance of recaptured image detectors** on raw, recaptured, and recaptured adversarial images of an alternate dataset [53] using a MacBook screen and iPhone. Even though our attack was not trained on this dataset, we observed a significant drop in accuracy for recaptured adversarial images.

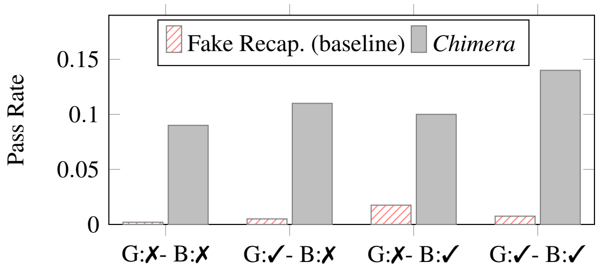
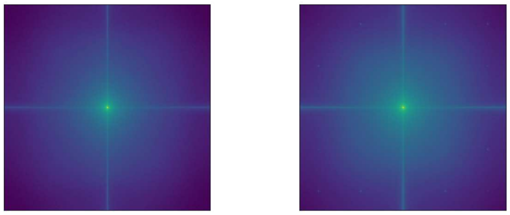


Figure 8: The portion of recaptured fake images (baseline) and recaptured adversarial fake images (*Chimera*) that **fool *both*** recaptured image and deepfake detection. For our evaluation, we select the better performing FatFormer [35] and TwoB_DCT [33] with all four training schemes - i.e., with and without blurring (B) and/or generalization (G).

## 6.5 Layered Defense Detection

We report the success rate of *Chimera* when both recapture and deepfake detectors are used - an attack is successful if it can bypass both detectors simultaneously.

Results are presented in Figure 8. We provide the findings for all four training models discussed in Section 6.2. As shown in the figure, *Chimera* significantly increases the success rate of the attack—from less than 1% to approximately 14% in the best case. The crucial takeaway from this experiment is that images generated by our system that successfully pass the tests (about 15%) will possess the following qualities: ***(i)*** they are labeled "real" when evaluated by state-of-the-art deepfake detectors, meaning they will not be identified by machine learning-based deepfake detection mechanisms; ***(ii)*** they are classified as "raw" when assessed by cutting-edge recapture detection methods; ***(iii)*** they include cryptographic signatures and are verified as genuine raw images taken by a real camera (adversarial recapture); and ***(iv)*** they visually resemble authentic raw images, as demonstrated in Figure 6.



(a) Raw Fake Images (b) Recaptured Adv Fake Images

Figure 9: Comparison of average frequency spectra between raw fake and recaptured Chimera adversarial fake images.

## 6.6 Additional Results

We report the detailed results for image recapture detection, including the breakdown of true/false positives in the Appendix. Furthermore, we present the results of an ablation study where adversarial training is used to make the recapture detectors robust against our attack. As can be seen in the Appendix (see Table 8), *Chimera* remains effective even when through adversarial training.

# 7 Discussion

## 7.1 Limitations

Due to the lossy nature of recapturing a picture and the inherent limitations of GANs, the inverse generator is unable to produce an image identical to the original raw image after recapture. The GAN's loss function is designed to minimize the difference between the generated and target data distributions, rather than learning a precise one-to-one mapping. As a result, while training the inverse generator, the simulator may overcompensate for certain features that it associates with the recapture process, causing inaccuracies in the target distribution of the inverse generator. This hints at the possibility that a detector can be built to reliably differentiate between raw images and recaptured adversarial images.

Additionally, GAN-generated artifacts were observed during experiments involving the iPhone. As a result, when recapturing adversarial images, we applied a slight zoom before taking the pictures to exclude these artifacts from the frame. This modification may account for the observed increase in FID scores in Table 2.

**Targeting Deepfake Detectors.** To directly target deepfake detectors in our pipeline, we explored two methods for crafting adversarial examples. The first approach involved modifying the attack training process by adding a term to the loss function that maximizes the loss of the deepfake detector. However, this method proved infeasible due to memory constraints during training, and even with additional memory, convergence issues may arise because the optimization landscape is highly non-convex and GAN training is unstable.

The second method involved producing adversarial images using our current pipeline, and then applying white-box Projected Gradient Descent (PGD) noise directly targeting the deepfake detector. Initial experiments with this approach showed that it did not significantly reduce the detection accuracy of the deepfake models. This may be since perturbations crafted digitally rely on precise pixel-level changes, but when an image is transferred to the physical domain, the loss of fine-grained details can reduce the effectiveness of the adversarial noise. We did not consider patch attacks, which are better suited for physical space scenarios, as they modify the content of the image; however, realistic and inconspicuous patches may still be a viable avenue for future exploration. Alternatively, techniques such as Expectation Over Transformation (EOT) [8] could be utilized to craft perturbations that are more resilient to physical transformations and thus can be incorporated into our attack pipeline.

## 7.2 Artifacts Produced by Chimera

From Figure 6, we can see that recaptured adversarial images produced by Chimera exhibit some artifacts that arise from both camera-side and display-side modifications. However, our experiments show that these artifacts are difficult to leverage as a defense against our attacks. For example, in Figure 9, we employ a frequency domain analysis [63] by calculating the average Fourier transform outputs for 500 raw fake images and 500 recaptured adversarial fake images and draw their average frequency spectra. The figure reveals that their average frequency spectra are highly similar, with only minor differences in the distribution of frequency components. The similarity arises because the inverse generator in Chimera is trained to minimize the focal frequency loss between raw and recaptured adversarial images.

Although some small differences in the frequency domain exist between the raw and recaptured images, they are difficult to take advantage of since they are unique to a particular camera-display setup that is not known a priori by the defender. As a result, defenders can not effectively exploit such artifacts for a defense. As shown in Table 8 in the Appendix, an adversarially trained recapture detector can defend against Chimera-produced images taken from the same screen used during adversarial training (AT) but fails to generalize to images taken from a different screen.

## 7.3 Countermeasures

**Hardware-based Defense.** Our attack is adapted to a particular hardware configuration where the capture device only has one camera. One of the primary reasons for the difficulty of distinguishing between “fake” and “real” images is due to the camera projection during the image formation process [48] which maps both images to a 2D plane. However, a more sophisticated capture device, such as one with multiple cameras or sensors such as LiDAR, opens the door to more effective defense measures that can circumvent our attack. Such capture devices can configure multiple cameras into a stereo setup [36] or use the LiDAR [34] to sense depth. With depth, a defender can then potentially identify the difference between a recaptured and real image since recaptured images are flat and 2D in nature. Overall, we believe that camera stacks with depth-sensing can effectively differentiate between real images and Chimera.

**Active Monitoring of Captured Images.** Our method requires capturing at least several hundred recaptured images to train the simulator and inverse generator. During this process, we are not actively attacking the system and these recaptures may be labeled as recaptured. A potential countermeasure to our method is to actively monitor all captures and detect if there is an abnormal number of recaptures taken that would allow for attacking the system with our method. For example, cameras integrated with active monitoring mechanisms can address this issue by detecting abnormal patterns of recapture attempts. However, this method would be obsolete with advances in training schemes that require a very small number of training images, which can be an avenue for future work.

## 8 Conclusions

This paper presented a novel attack methodology that exposes critical vulnerabilities in image-based cryptographic signing methods, image recapture detectors, and even Deepfake detectors. Our method’s strength manifests in its ability to be hardware-aware and compensate or personalize to the details of any hardware setup of camera and display. This versatility along with its ability to greatly lower the accuracy of detectors while maintaining high perceptual quality highlights the need for more advanced and resilient detection mechanisms. As digital forensics evolves, the results of this work highlight the importance of developing countermeasures for the integrity of our digital media in an increasingly adversarial environment.

**Ethics Statement:** We strongly condemn any misuse of the methods outlined in this work for creating deceptive content, and we emphasize that its purpose is solely to inform and enhance detection and defense strategies. Without this scheme, existing cryptographic solutions for deepfake detection provide a false sense of security.

**Open Science Policy.** We have open-sourced our tool at: `https://github.com/ssysarch/Chimera`.

## Acknowledgement

We thank our shepherd for their guidance. This work has been supported, in part, by CISCO Systems Inc., NSF grants

2046737, 2211301, 2303115, 2312089, 2323105, 2317184, and 2411153, a DARPA Young Faculty Award (Kadambi), an Army Young Investigator Award (Kadambi), and a gift received from Accenture. The views and findings in this paper are those of the authors and do not necessarily reflect the views of Cisco, NSF, DARPA, Army, and Accenture.

## References

[1] C2PA: An open technical standard providing publishers, creators, and consumers the ability to trace the origin of different types of media. http://https://c2pa.org/. Accessed: 2024-02-01.

[2] Nikon, sony and canon fight ai fakes with new camera tech. https://asia.nikkei.com/Business/Technology/Nikon-Sony-and-Canon\-fight-AI-fakes-with-new-camera-tech. Accessed: 2024-08-29.

[3] Sifat Muhammad Abdullah, Aravind Cheruvu, Shravya Kanchi, Taejoong Chung, Peng Gao, Murtuza Jadliwala, and Bimal Viswanath. An analysis of recent advances in deepfake image detection in an evolving threat landscape. In *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 2024.

[4] E. Abraham. Moiré pattern detection using wavelet decomposition and convolutional neural network. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1275–1279, Nov 2018.

[5] Shruti Agarwal, Wei Fan, and Hany Farid. A diverse large-scale dataset for evaluating rebroadcast attacks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1997–2001. IEEE, 2018.

[6] Isaac Amidror. *The Theory of the Moiré Phenomenon: Volume I: Periodic Layers*, volume 38. Springer Science & Business Media, 2009.

[7] Areesha Anjum and Saiful Islam. Recapture detection technique based on edge-types by analysing high-frequency components in digital images acquired through lcd screens. *Multimedia tools and applications*, 79(11):6965–6985, 2020.

[8] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pages 284–293. PMLR, 2018.

[9] Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. Recent advances in adversarial training for adversarial robustness. *arXiv preprint arXiv:2102.01356*, 2021.

[10] Dan Boneh, Andrew J Grotto, Patrick McDaniel, and Nicolas Papernot. How relevant is the turing test in the age of sophisbots? *IEEE Security & Privacy*, 17(6):64–71, 2019.

[11] Hong Cao and Alex C Kot. Identification of recaptured photographs on lcd screens. In *2010 IEEE International conference on acoustics, speech and signal processing*, pages 1790–1793. IEEE, 2010.

[12] Christopher Chun Ki Chan, Vimal Kumar, Steven Delaney, and Munkhjargal Gochoo. Combating deepfakes: Multi-lstm and blockchain as proof of authenticity for digital media. In *2020 IEEE/ITU International Conference on Artificial Intelligence for Good*. IEEE, 2020.

[13] Yushi Cheng, Xiaoyu Ji, Lixu Wang, Qi Pang, Yi-Chao Chen, and Wenyuan Xu. mID: Tracing screen photos via {Moiré} patterns. In *30th USENIX Security Symposium (USENIX Security 21)*, 2021.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[15] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[16] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[17] Wei Fan, Shruti Agarwal, and Hany Farid. Rebroadcast attacks: Defenses, reattacks, and redefenses. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 942–946. IEEE, 2018.

[18] Maayan Frid-Adar, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. Synthetic data augmentation using gan for improved liver lesion classification. In *2018 IEEE 15th international symposium on biomedical imaging*. IEEE, 2018.

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[20] Guardian. How did donald trump end up posting taylor swift deepfakes?. https://www.theguardian.com/technology/article/2024/aug/24/trump-taylor-swift-deepfakes-ai. Accessed: 2024-08-29.

[21] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[23] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[24] Liming Jiang, Bo Dai, Wayne Wu, and Chen Change Loy. Focal frequency loss for image reconstruction and synthesis. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2021.

[25] James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. Pate-gan: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations*, 2018.

[26] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

[27] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019.

[28] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020.

[29] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[30] Binh M Le, Jiwon Kim, Shahroz Tariq, Kristen Moore, Alsharif Abuadbba, and Simon S Woo. Sok: Facial deepfake detectors. *arXiv preprint arXiv:2401.04364*, 2024.

[31] Changjiang Li, Li Wang, Shouling Ji, Xuhong Zhang, Zhaohan Xi, Shanqing Guo, and Ting Wang. Seeing is living? rethinking the security of facial liveness verification in the deepfake era. In *31st USENIX Security Symposium (USENIX Security 22)*, 2022.

[32] Haoliang Li, Shiqi Wang, and Alex C Kot. Image recapture detection with convolutional and recurrent neural networks. *Electronic Imaging*, 29:87–91, 2017.

[33] Jiaxing Li, Chenqi Kong, Shiqi Wang, and Haoliang Li. Two-branch multi-scale deep neural network for generalized document recapture attack detection. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023.

[34] You Li and Javier Ibanez-Guzman. Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems. *IEEE Signal Processing Magazine*, 37(4):50–61, 2020.

[35] Huan Liu, Zichang Tan, Chuangchuang Tan, Yunchao Wei, Jingdong Wang, and Yao Zhao. Forgery-aware adaptive transformer for generalizable synthetic image detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10770–10780, 2024.

[36] Yi Ma, Stefano Soatto, Jana Košecká, and Shankar Sastry. *An invitation to 3-d vision: from images to geometric models*, volume 26. Springer, 2004.

[37] Jaron Mink, Licheng Luo, Natã M Barbosa, Olivia Figueira, Yang Wang, and Gang Wang. {DeepPhish}: Understanding user trust towards artificially generated profiles in online social networks. In *31st USENIX Security Symposium (USENIX Security 22)*, 2022.

[38] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24480–24489, 2023.

[39] Donie O'Sullivan. Inside the pentagon's race against deepfake videos. *CNN*, 2020.

[40] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *European Conference on Computer Vision*, pages 86–103, 2020.

[41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[42] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[43] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[44] Leonard Rosenthol. C2pa: the world's first industry standard for content provenance. In *Applications of Digital Image Processing XLV*. SPIE, 2022.

[45] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35, 2022.

[46] Zeyang Sha, Zheng Li, Ning Yu, and Yang Zhang. Defake: Detection and attribution of fake images generated by text-to-image generation models. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 3418–3432, 2023.

[47] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1), 2014.

[48] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022.

[49] Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, and Yunchao Wei. Learning on gradients: Generalized artifacts representation for gan-generated images detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12105–12114, 2023.

[50] Razaib Tariq, Minji Heo, Simon S. Woo, and Shahroz Tariq. Beyond the screen: Evaluating deepfake detectors under moire pattern effects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2024.

[51] Thirapiroon Thongkamwitoon, Hani Muammar, and Pier-Luigi Dragotti. An image recapture detection algorithm based on learning dictionaries of edge profiles. *IEEE Transactions on Information Forensics and Security*, 10(5):953–968, 2015.

[52] Nitasha Tiku. Ai can now create any image in seconds, bringing wonder and danger. *The Washington Post*, 2022.

[53] Bojan Tunguz. 1 million fake faces. https://www.kaggle.com/datasets/tunguz/1-million-fake-faces/data. Accessed: 2024-12-28.

[54] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[55] Verge. Liveness tests used by banks to verify id are "extremely vulnerable" to deepfake attacks. *The Verge*, 2022.

[56] Alexander Vilesov, Yuan Tian, Nader Sehatbakhsh, and Achuta Kadambi. Solutions to deepfakes: Can camera hardware, cryptography, and deep learning verify real images? *arXiv preprint arXiv:2407.04169*, 2024.

[57] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8695–8704, 2020.

[58] Pengpeng Yang, Rongrong Ni, and Yao Zhao. Recapture image forensics based on laplacian convolutional neural networks. In *Digital Forensics and Watermarking: 15th International Workshop, IWDW 2016, Beijing, China, September 17-19, 2016, Revised Selected Papers 15*, pages 119–128. Springer, 2017.

[59] Abbas Yazdinejad, Reza M. Parizi, Gautam Srivastava, and Ali Dehghantanha. Making sense of blockchain for ai deepfakes technology. In *2020 IEEE Globecom Workshops (GC Wkshps*, pages 1–6, 2020.

[60] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[61] Ning Yu, Larry S Davis, and Mario Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7556–7566, 2019.

[62] Huanjing Yue, Yijia Cheng, Xin Liu, and Jingyu Yang. Recaptured raw screen image and video demoir\'eing via channel and spatial modulations. *arXiv preprint arXiv:2310.20332*, 2023.

[63] Xu Zhang, Svebor Karaman, and Shih-Fu Chang. Detecting and simulating artifacts in gan fake images. In *2019 IEEE international workshop on information forensics and security (WIFS)*, pages 1–6. IEEE, 2019.

[64] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

|  | Finetuned? | Blur? | Raw Real | Raw Fake | Recap Real | Recap Fake | Adv. Recap Real | Adv. Recap Fake |
|---|---|---|---|---|---|---|---|---|
| Twob_DCT [33] | Yes | Yes | 0.918 | 0.723 | 0.868 | 0.995 | 0.595 | 0.638 |
|  | Yes | No | 0.895 | 0.780 | 0.930 | 0.972 | 0.612 | 0.688 |
|  | No | Yes | 0.837 | 0.848 | 0.922 | 0.897 | 0.513 | 0.520 |
|  | No | No | 0.827 | 0.903 | 0.968 | 0.963 | 0.665 | 0.472 |
| Twob_DWT [33] | Yes | Yes | 0.873 | 0.640 | 0.858 | 0.985 | 0.577 | 0.570 |
|  | Yes | No | 0.862 | 0.758 | 0.980 | 1.000 | 0.705 | 0.728 |
|  | No | Yes | 0.867 | 0.915 | 0.952 | 0.973 | 0.418 | 0.335 |
|  | No | No | 0.867 | 0.930 | 0.942 | 0.952 | 0.418 | 0.283 |
| MoireDet [4] | Yes | Yes | 0.962 | 0.878 | 0.823 | 0.988 | 0.272 | 0.413 |
|  | Yes | No | 0.923 | 0.920 | 0.967 | 0.995 | 0.368 | 0.363 |
|  | No | Yes | 0.925 | 0.988 | 0.922 | 0.957 | 0.130 | 0.028 |
|  | No | No | 0.922 | 0.993 | 0.845 | 0.810 | 0.168 | 0.045 |

Table 6: Performance of recaptured image detectors on raw, recaptured, and recaptured adversarial images using a MacBook screen and iPhone. All models exhibit high accuracy for raw and recaptured images, as images from the setup were included in the training set. However, we observe a significant drop in accuracy for recaptured adversarial images, indicating that the models are susceptible to our attack

# 9 Appendix

## 9.1 Detailed Results for Recapture Detection

Tables 6 and 7 show the results of image recapture detection on two different displays. The true positive and false negative breakdowns are shown in Table 9. Note that the increase in false negative rates indicates the success of the attack (i.e., the opposite of benign).

## 9.2 Ablation Study: Adversarial Training

If the inverse generator $G^*_{inv}$ could perfectly invert the recapture function $g$, it would be impossible to differentiate between raw and recaptured adversarial images. However, this is not the case, due to the limitations of our method as seen with the poor quality of the last sample in Figure 6. Therefore, we explore the impact of adversarial training in mitigating the effectiveness of our attack strategy. Because our setup differs from traditional adversarial attacks, instead of utilizing an adversarial training framework like TRADES [9], we simply augment the training dataset of our detector by including recaptured adversarial images, specifically those from the MacBook screen. As outlined in Table 8, the attack success rate against the detectors trained with adversarial samples is much lower for the attack carried out using the MacBook screen. However, adversarial training does not seem to provide the same benefit for unseen screens and their attacks, as seen in the low accuracies for the second screen, indicating that for adversarial training to be a feasible solution, images from a very wide range of screens and respective adversarial samples must be collected as the training set.

|  | Finetuned? | Blur? | Recap Real | Recap Fake | Adv. Recap Real | Adv. Recap Fake |
|---|---|---|---|---|---|---|
| Twob_DCT [33] | Yes | Yes | 0.777 | 0.920 | 0.495 | 0.620 |
|  | Yes | No | 0.922 | 0.912 | 0.737 | 0.822 |
|  | No | Yes | 0.440 | 0.398 | 0.372 | 0.290 |
|  | No | No | 0.830 | 0.780 | 0.740 | 0.750 |
| Twob_DWT [33] | Yes | Yes | 0.738 | 0.890 | 0.415 | 0.610 |
|  | Yes | No | 0.960 | 0.982 | 0.702 | 0.803 |
|  | No | Yes | 0.590 | 0.540 | 0.298 | 0.210 |
|  | No | No | 0.793 | 0.788 | 0.637 | 0.597 |
| MoireDet [4] | Yes | Yes | 0.485 | 0.677 | 0.177 | 0.475 |
|  | Yes | No | 0.790 | 0.913 | 0.423 | 0.575 |
|  | No | Yes | 0.485 | 0.522 | 0.362 | 0.250 |
|  | No | No | 0.495 | 0.260 | 0.742 | 0.682 |

Table 7: Performance of recaptured image detectors on raw, recaptured, and recaptured adversarial images using a Monitor and iPhone. The detectors perform worse across the board for this setup compared to Table 6, with the models trained without blur augmentations performing better.

|  | Raw |  | Recap 1 |  | Adv 1 |  | Recap 2 |  | Adv 2 |  |
|---|---|---|---|---|---|---|---|---|---|---|
|  | Real | Fake | Real | Fake | Real | Fake | Real | Fake | Real | Fake |
| DWT (No G, No Blur) | 0.822 | 0.658 | 0.893 | 0.978 | 0.898 | 0.965 | 0.802 | 0.855 | 0.507 | 0.617 |
| DWT (No G, Blur) | 0.937 | 0.602 | 0.635 | 0.863 | 0.798 | 0.928 | 0.378 | 0.560 | 0.198 | 0.437 |
| DCT (No G, No Blur) | 0.888 | 0.707 | 0.923 | 0.950 | 0.900 | 0.970 | 0.768 | 0.817 | 0.620 | 0.790 |
| DCT (No G, Blur) | 0.948 | 0.668 | 0.665 | 0.865 | 0.808 | 0.962 | 0.340 | 0.550 | 0.262 | 0.488 |
| DCT (G, No Blur) | 0.898 | 0.955 | 0.808 | 0.683 | 0.475 | 0.258 | 0.495 | 0.373 | 0.508 | 0.385 |
| DCT (G, Blur) | 0.875 | 0.895 | 0.877 | 0.875 | 0.642 | 0.518 | 0.567 | 0.485 | 0.388 | 0.302 |

Table 8: Adversarial Training Results: Performance of different models on various image types. The models are evaluated on images from Screen 1 (i.e. MacBook), which was included in the training set, and Screen 2 (i.e. Monitor), which was not included in the training set. Adversarially generated images (Adv 1 and Adv 2 for the recaptured adversarial images on Screen 1 and 2, respectively) are also included to assess the impact of adversarial training.

| Name | Finetuned? | Blur? | TP (benign) | FN (benign) | TP (adv) | FN (adv) |
|---|---|---|---|---|---|---|
| dct | Yes | Yes | 0.9315 | 0.0685 | 0.6165 | 0.3835 |
| dct | Yes | No | 0.95 | 0.05 | 0.65 | 0.35 |
| dct | No | Yes | 0.9095 | 0.0905 | 0.5165 | 0.4835 |
| dct | No | No | 0.9655 | 0.0345 | 0.5685 | 0.4315 |
| dwt | Yes | Yes | 0.9215 | 0.0785 | 0.5735 | 0.4265 |
| dwt | Yes | No | 0.99 | 0.01 | 0.7165 | 0.2835 |
| dwt | No | Yes | 0.9625 | 0.0375 | 0.3765 | 0.6235 |
| dwt | No | No | 0.947 | 0.053 | 0.3505 | 0.6495 |
| md | Yes | Yes | 0.9055 | 0.0945 | 0.3425 | 0.6575 |
| md | Yes | No | 0.981 | 0.019 | 0.3655 | 0.6345 |
| md | No | Yes | 0.9395 | 0.0605 | 0.079 | 0.921 |
| md | No | No | 0.8275 | 0.1725 | 0.1065 | 0.8935 |

Table 9: True positive (TP) and false negative (FN) breakdowns for different configurations. *Chimera* decreases the true positive rate and increases the false negative rate (i.e., the attack is successful).