TrustMark: Robust Watermarking and Watermark Removal for Arbitrary Resolution Images

Tu Bui[1] Shruti Agarwal[2] John Collomosse[1,2]
[1]University of Surrey, UK.
[2]Adobe Research, USA.
tuvbui90@gmail.com, {shragarw,collomos}@adobe.com

Abstract

Imperceptible digital watermarking is important in copyright protection, misinformation prevention, and responsible generative AI. We propose TrustMark - a watermarking method that leverages a spatio-spectral loss function and a 1×1 convolution layer to enhance encoding quality. TrustMark is robust against both in-place and out-of-place perturbations while maintaining image quality above 43 dB. Additionally, we propose ReMark, a watermark removal method designed for re-watermarking, along with a simple yet effective algorithm that enables both TrustMark and ReMark to operate across arbitrary resolutions. Our methods achieve state-of-art performance on 3 benchmarks[1].

1. Introduction

Advances in generative AI (GenAI) present fresh challenges in combating misinformation and identifying the origins (provenance) of images. Emerging content provenance standards (*e.g*. C2PA [16]) address this challenge by embedding metadata within images to describe how they were created. However, such metadata is often removed by non-compliant platforms (*e.g*. social media) as images are posted and shared. Digital watermarking offers a way to embed an imperceptible identifier that may be used to recover provenance information in this situation [19]. In this paper we propose TrustMark, a novel image watermarking model that can be applied to general images (photos or GenAI) in order to help identify their provenance.

The goal of (imperceptible) image watermarking is to embed a message (here, provenance data) directly within the image content in such a way that the changes to the image are imperceptible yet detectable by a 'watermark decoder'. TrustMark is designed to address several requirements specific to image provenance. First, provenance is a non-steganographic use case for watermarking, where public detection and decoding is required, not conditional on any secret key. For example, a web browser may detect a watermark, and so present provenance data to enable users to make more informed trust decisions about an image. Second, creative tools may (re-)encode these openly detectable identifiers in images as they are edited. Third, a key consideration for creative practice is visual quality degradation due to watermarking, and 're-watermarking' (watermark removal and replacement). Creative imagery is often high resolution, and the watermark should be imperceptible. Finally, the watermark should robust to non-editorial transformations (renditions) performed by content platforms such as social media. Many robust watermarks operate only over fixed resolutions [13, 40, 48, 66, 77]. We propose TrustMark to address these challenges. Our contributions are:

1. State-of-art watermarking performance. We extend the existing encoder-decoder based watermarking methods [66, 73, 77] with a new backbone, a post-process layer and a novel frequency-based (FFL) loss for improved preservation of high frequency detail in the watermarked image. Robustness of the encoding is encouraged via extensive noise simulation during training. TrustMark achieves state-of-art performance in both imperceptibility and watermark recovery on three benchmarks.

2. Watermark removal network ReMark, to restore the original image with high quality, useful for applications such as re-watermarking.

3. Resolution Scaling method to extend TrustMark, ReMark and other watermarking methods to operate over images of arbitrary resolution.

TrustMark thus addresses practical challenges across the watermark lifecycle for creative work: imperceptibility and robustness; scaling for arbitrary resolution; and ReMark enabling high quality restoration for re-watermarking.

2. Related work

Media provenance is the focus of cross-industry coalitions (*e.g*. CAI [60], Origin [3]) and emerging standards (*e.g*. C2PA [16]) that encode a metadata 'manifest' within an image, containing information on its origins. When that metadata is stripped, a perceptual hash [5, 6, 52] can be used to lookup the manifest in a database or distributed ledger *e.g*. blockchain [8, 9]. Yet reliance on near-duplicate search is inexact, and provides no signal to trigger such a lookup.

[1]Models and code are released under MIT license at https://github.com/adobe/trustmark.

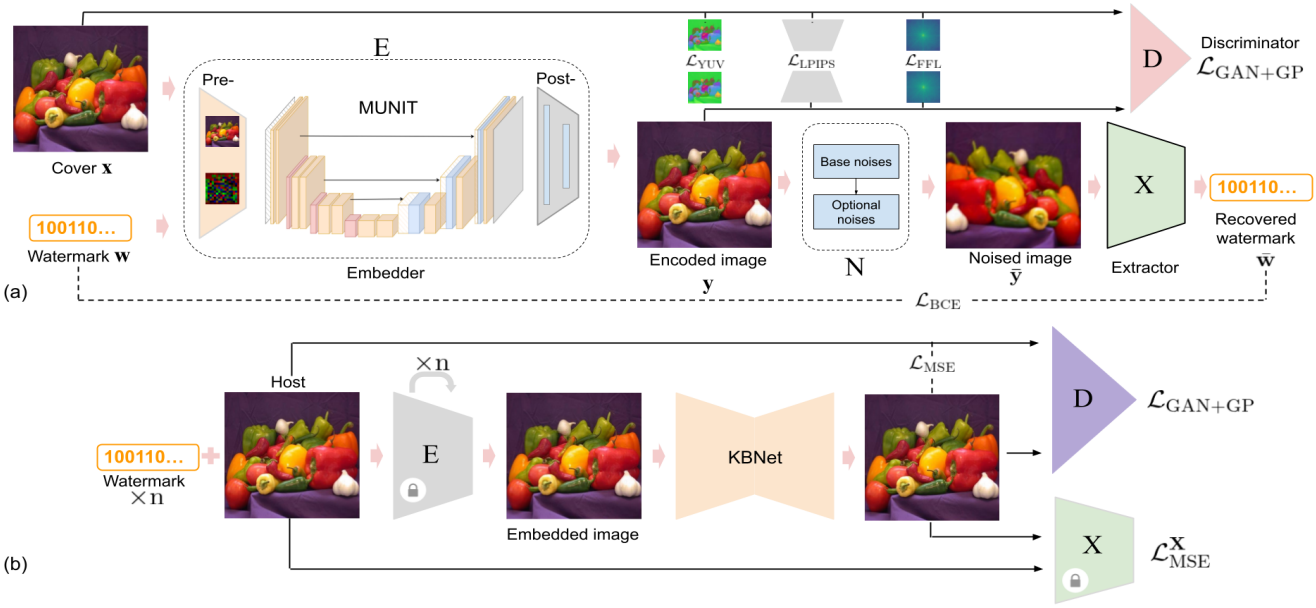


Figure 1. Proposed architecture of TrustMark (a). The embedder **E** encodes a watermark into a cover image robustly using a noise module **N** to simulate common perturbations on the encoded image. The extractor **X** recovers the watermark from the encoded image. The ReMark network (b) removes the watermark to enable re-watermarking of the image.

(Re-)watermarking provides an alternative – to insert and update an imperceptible identifier to recover stripped metadata that 'good' actors may insert to optionally reinforce the authenticity of their content. Given the opt-in nature of provenance, we do not consider stripping of provenance (or watermark) to be a concern as this reduces rather than raises authenticity of content. Indeed it may be a useful step in enabling re-watermarking of content to update provenance data as an image is editted.

Classical Watermarking explored Least Significant Bit (LSB) [69] to embed a secret in the lowest order bits of each pixel, producing images perceptually indistinguishable from the original ('cover') image. Since then, several techniques have exploited the spatial [29, 47, 64] and frequency [34, 35, 43, 51, 56, 58] domains to embed the secret. A classical watermark of relevance is dwtDctSvd [51] often used to watermark Stable Diffusion generative AI images. Stirmark [54] was a classical benchmark important in driving early advances in robustness using noise, filtering and geometric transformations. More recently evaluations added colour and other non-geometric perturbations in addition to these kinds of transformations, and we evaluate similarly for TrustMark.

Deep Watermarking has been shown to provide robustness to noises while maintaining good quality of the generated image [68]. HiDDeN [77] was the first end-to-end trained watermarking network that used the encoder-decoder architecture for watermark embedding. This was followed by several later works [14, 21, 50, 66, 70, 73] with great improvement in embedding quality and robustness. These works mostly encode secrets and covers jointly, often with an UNet-like model and skip connections to preserve small details in the cover images [21, 66, 73]. Notably, StegaStamp [66] incorporates spatial transformers for robustness against geometric transformations. RivaGAN [73] employs attention mechanism for video watermarking. SSL [25] takes a different approach, watermarking images in the latent space at inference time via back-propagation, achieving superior imperceptibility score at cost of speed. RoSteALS [13] also proposes to watermark via the latent code of a frozen VQVAE [22], achieving state-of-art robustness however its imperceptibility is limited by VQVAE [22] reconstruction quality. Recently region watermarking has been explored for localized embedding [61] or manipulation detection [74]. Other recent techniques train GenAI models to embed watermarks during generation [2, 24, 26, 72], which are less general than TrustMark which, as a 'post-hoc' approach, may be applied to any image.

Watermark removal has been investigated in the context of inpainting *visible* watermarks [15, 20, 49, 53] but *invisible* watermark removal and its application in 're-watermarking' are not yet explored. We show that naive watermark removal (using adversarial attacks) and re-watermarking (simply applying new watermarks on a watermarked image) methods worsen image quality (Sec. 4.4). A recent work by Zhao *et al*. [76] leverages random noise to destroy the watermark followed by generative AIs to recover the image. We show that [76] struggles to work on robust watermarking models such as TrustMark; also images recovered by generative AIs tend to visually differ from the originals (also reported in [13]). In contrast, our ReMark significantly improves image quality and is useful even after multiple watermarking times.

Arbitrary resolution is not a problem for shallow water-

marking methods [51] since they mostly operate on frequency spectrum. On the other hand, the rigid encoder-decoder architecture of deep methods constrains the embedding at a fixed resolution that the network is designed for *e.g*. HiDDeN [77] operates at 128×128, RoSteALS [13] at 256×256, StegaStamp [66] at 400×400. RivaGAN [73] works on arbitrary resolution because their encoder maintains the image spatial resolution during watermarking *i.e*. no bottleneck layer. In contrast, we propose Resolution Scaling as a flexible post process that can be applied to any fixed resolution watermarking methods.

3. Methodology

We describe our watermarking network (TrustMark) and the watermark removal network (ReMark), both operate on 256×256 fixed-resolution images, in Sec. 3.1 and Sec. 3.2 respectively; and outline how the two can be adapted to work on arbitrary resolution images at inference time in Sec. 3.3.

3.1. Watermarking network

Inspired by HiDDEN [77] and StegaStamp [66], TrustMark (Fig. 1a) also comprises an embedder module $\mathbf{E}$ to encode the watermark into a host ('cover') image, an extractor module $\mathbf{X}$ to recover the watermark from that image, and a noise module $\mathbf{N}$ to perturb the image during training. We describe each component and the difference below.

3.1.1. Watermark Embedder

The embedder network $\mathbf{E}$ first accepts a cover image $\mathbf{x} \in \mathbb{R}^{256\times256\times3}$ and a watermark $\boldsymbol{w} \in \{0,1\}^l$ (with l being the watermark size) into its pre-processing module $\mathbf{E}_{\text{pre}}(\mathbf{x}, \boldsymbol{w}) \in \mathbb{R}^{256\times256\times d}$, where d is the internal feature dimension. Following [4, 66], we design $\mathbf{E}_{\text{pre}}$ as an early image-watermark fusion network – the watermark is interpolated to match the cover image's dimension, then the concatenated cover-watermark feature map is convolved with d 3×3 filters to make a d-channel image output. Differently, we employ a MUNIT-based [38] network originally designed for style transfer as our watermark backbone. Specifically, we remove channel normalization, double the network depth but halve the internal dimension, effectively reducing the model size 2x. Finally, a post-process module converts MUNIT's n-channel output back to the RGB space, $\bar{\mathbf{x}} = \mathbf{E}_{\text{post}}(.) \in \mathbb{R}^{256\times256\times3}$. While this could be implemented using just a single convolutional layer [4, 66, 77], we found that a more complex $\mathbf{E}_{\text{post}}$ is needed to retain high frequency details in the encoded image. We therefore leverage multiple 1×1 convolution layers which act as channel-wise pooling layers commonly used in dimensionality reduction and feature learning networks [44, 63]. These 1×1 convolution layers are separated by SiLU and end with a $tanh()$ activation to constraint the output pixel values to range $[-1, 1]$. Note that our embeder $\mathbf{E}$ outputs the encoded image directly, $\mathbf{y} = \mathbf{E}(\mathbf{x}, \boldsymbol{w})$, instead of estimating the residual artifacts to be added to the cover image as in StegaStamp [66] or RivaGAN [73].

Algorithm 1: Resolution scaling for watermark embedding and removal.

Input: Input image $\mathbf{x}$, trade-off factor $\lambda > 0$, [binary watermark vector $\boldsymbol{w}$]
Output: Restored image $\mathbf{y}$
Data: Embedding network $\mathbf{E}$ or Removal network $\mathbf{R}$

H, W := $\mathbf{x}$.height, $\mathbf{x}$.width
$\mathbf{x} \leftarrow \mathbf{x}/127.5 - 1$ `// Normalize to range [-1,1]`
$\bar{\mathbf{x}} := \text{interpolate}(\mathbf{x}, (256, 256))$
if *model is watermarking* **then**
  $\mathbf{r} := \mathbf{E}(\bar{\mathbf{x}}, \boldsymbol{w}) - \bar{\mathbf{x}}$ `// residual image`
else
  $\mathbf{r} := R(\bar{\mathbf{x}}) - \bar{\mathbf{x}}$
$\mathbf{r} \leftarrow \text{interpolate}(\mathbf{r}, (\text{H}, \text{W}))$
$\mathbf{y} \leftarrow \text{clamp}(\mathbf{x} + \lambda\mathbf{r}, -1, 1)$ `// trade-off control`
$\mathbf{y} \leftarrow \mathbf{y} * 127.5 + 127.5$

3.1.2. Watermark extractor

The extractor network $\mathbf{X}$ aims to decode the watermark from the encoded image, $\bar{\boldsymbol{w}} = \mathbf{X}(\mathbf{y}) \in \{0,1\}^l$. This is challenging as the watermark signal is perceptually invisible. We observe that only resnet-based networks supervised under a particular training scheme fit the task (c.f. Sec. 4.1 and Sec. 4.5). Here we employ the standard ResNet50 [32] with the last layer being replaced by a l-dimension sigmoid-activated FC to predict the l-bit watermark.

3.1.3. Noise model

Robustness to noises is an important factor for invisible watermarking. To expose the extractor $\mathbf{X}$ to various noise sources when it is being jointly trained with the embedder $\mathbf{E}$, we insert a noise model $\mathbf{N}$ after the encoded image - $\tilde{\mathbf{y}} = \mathbf{N}(\mathbf{y})$. $\mathbf{N}$ consists of 3 geometrical transformations (random flip, crop and resize) and 15 perturbation sources (random JPEG compression, brightness, hue, contrast, sharpness, color jitter, RGB shift, saturation, grayscale, Gaussian blur, median blur, box blur, motion blur, Gaussian noise, posterize). Each encoded image is perturbed with 3 geometrical transformations ('base transforms' in Fig. 1) and 2 other random noises ('optional transforms'). All 18 transforms are differentiable so that errors can be propagated back to the embedder. More details are in the Sup.Mat.

3.1.4. Losses

Overall, training TrustMark involves balancing image quality (via $\mathbf{E}$) with watermark recovery (via $\mathbf{X}$) in the presence of complex noise simulation.

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{\text{quality}}(\mathbf{x}, \mathbf{y}) + \mathcal{L}_{\text{recovery}}(\boldsymbol{w}, \bar{\boldsymbol{w}}) \quad (1)$$

where α is the trade-off hyper-parameter. We adopt RoSteALS [13] strategy to start training with a low value of α to prioritize watermark recovery then linearly increase to a threshold α_{max}, which can be set prior training to exert TrustMark's controlability (see Sec. 4.2).

$\mathcal{L}_{\text{recovery}}(\boldsymbol{w}, \bar{\boldsymbol{w}})$ is the standard binary cross-entropy loss to bring the recovered watermark close to the original. The quality loss is defined as,

$$\mathcal{L}_{\text{quality}}(\mathbf{x}, \mathbf{y}) = \beta_{\text{YUV}}\mathcal{L}_{\text{YUV}} + \beta_{\text{LPIPS}}\mathcal{L}_{\text{LPIPS}} \quad (2)$$

$$+ \beta_{\text{FFL}}\mathcal{L}_{\text{FFL}} + \beta_{\text{GAN}}\mathcal{L}_{\text{GAN+GP}} \quad (3)$$

where $\beta_{\text{YUV}}, \beta_{\text{LPIPS}}, \beta_{\text{FFL}}$ and β_{GAN} are the weights of 4 loss terms. $\mathcal{L}_{\text{YUV}}(\mathbf{x}, \mathbf{y})$ is the mean squared error loss in the YUV pixel space, $\mathcal{L}_{\text{LPIPS}}(\mathbf{x}, \mathbf{y})$ is the perceptual loss following [13, 66]. Additionally, TrustMark is also trained in generative adversarial fashion with GAN loss,

$$\mathcal{L}_{\text{GAN+GP}}(\mathbf{x}, \mathbf{y}) = \mathop{\mathbb{E}}_{\mathbf{y}\sim\mathbb{P}_E}[\mathbf{D}(\mathbf{y})] - \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathbb{P}_{real}}[\mathbf{D}(\mathbf{x})] \quad (4)$$

$$+ \lambda \mathop{\mathbb{E}}_{\mathbf{y}\sim\mathbb{P}_E}[(||\nabla_y \mathbf{D}(\mathbf{y})||_2 - 1)^2] \quad (5)$$

where $\mathbf{D}$ is a discriminator to distinguish the encoded image from the original, λ is the gradient penalty loss weight for training stablization [31].

Finally, we propose to add a focal frequency loss (FFL) to bridge the gap between the cover and encoded image in frequency domain,

$$\mathcal{L}_{\text{FFL}}(\mathbf{x}, \mathbf{y}) = \rho_{f(\mathbf{x}),f(\mathbf{y})} \, ||f(\mathbf{y}) - f(\mathbf{x})||_2 \quad (6)$$

where $f(.)$ is the 2D Fourier transform function and $\rho_{f(\mathbf{x}),f(\mathbf{y})} \in \mathbb{R}^{256\times256\times3}$ is a dynamic weight matrix to balance the loss across frequency spectrum. It is reported in [10, 28] that synthezied images often exhibit artifacts in the frequency domain that can be easily spot by deep networks. FFL was first introduced in [41, 45] to reduce such artifacts. Here we leverage FFL to encourage higher watermark embedding quality instead (see Sec. 4.5 and Sup.Mat).

3.2. Watermark removal

Since the watermark is embedded at the high-frequency bands of the image, it can be removed by adding enough noises [76] or via adversarial attacks or even embedding a different watermark on top. However, such methods risk reducing the image quality. Here, we aim to design a reliable watermark removal network to not only remove the watermark but also enhance image quality (instead of worsening it). Assume access to the watermarking model, we treat watermark removal as a denoising task and integrate both the TrustMark's embedder and extractor to help regulate the training (Fig. 1(b)). Our proposed network, ReMark, has the backbone based on KBNet [75] – the current state-of-art model for image restoration. For each training cover image, we embed n random watermarks using $\mathbf{E}(.)$ to synthesize the input sample for the denoise model. Different from KBNet [75], ReMark is regulated to be close to the cover image using a combination of 3 losses: (i) a pixel loss $\mathcal{L}_{\text{MSE}}$ prioritizing Peak-Signal-To-Noise (PSNR) ratio directly, (ii) a discriminator loss $\mathcal{L}_{\text{GAN+GP}}$ similar to TrustMark, and (iii) watermark similarity loss $\mathcal{L}^{\mathbf{X}}_{\text{MSE}}$ to have the output image the same response to $\mathbf{X}(.)$ as the cover (close to random chance).

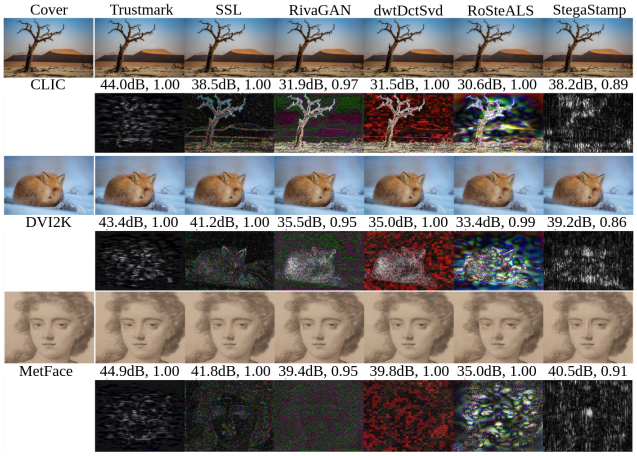


Figure 2. Representative watermarking examples for TrustMark and 5 baseline methods (SSL [25], RivaGAN [73], dwtDctSvd [51], RoSteALS [13], StegaStamp [66]) over 3 benchmarks (CLIC [67], DIV2k [1]) and MetFace [42]), using the same random watermark. The residual is amplified 20 times for visualization purpose.

3.3. Resolution Scaling

Working on arbitrary resolution is a difficult challenge for deep watermarking. Many existing methods only support a fixed resolution [13, 40, 66, 77]. Those supporting arbitrary resolutions require a special design in the model architecture *e.g.* [73]. Here, we propose Resolution Scaling as a post-hoc approach suitable for any watermarking model (Algorithm 1). First, the cover image of resolution H×W is downsampled to 256×256 upon which the watermark is embedded. Next, we compute the residual between the watermarked image and cover image at the embedding resolution. The residual is then rescaled back to H×W before being added to the original cover image (with control factor λ). Our method offers 4 advantages: (i) it executes at inference time and treats the model as a blackbox, therefore can be applied for any watermarking/removal algorithms; (ii) by interpolating only the residual, the high resolution output image can be derived directly from the original input (Line#9 of Algorithm 1) therefore maximizes image quality; (iii) the imperceptibility and robustness trade-off can be controlled at inference time via λ (increasing λ improves watermark recovery but also makes the watermark artifacts more visible); and (iv) our method is fast since image interpolation is the only overhead. While our Resolution Scaling is extremely simple, its underlining motivation is not trivial – the burden caused by arbitrary resolution is shifted from the embedder (remaining intact) to the extractor (needs to be robust to handle distorted residual after interpolation). For ReMark, artifact removal at 256×256 resolution also retains its effect after Resolution Scaling is applied (Sec. 4.4).

4. Experiments

4.1. Datasets, training details, and baselines

Datasets. We follow the same settings of [13] to train our models on 101K images from the MIRFlickR 1M dataset

[39] (100K images for training and 1K for validation) and evaluate on the CLIC [67] and MetFace [42] benchmarks. Additionally, we evaluate on DIV2K [1] – a more diverse and higher quality benchmark popular for super-resolution testing. Since the DIV2K test set is not visible to public, we use both the training and validation images for testing. Unless otherwise specified, all experiments are evaluated on DIV2K. At test time, every image is associated with a random watermark and the encoded image is perturbed with random noises described in Sec. 3.1.

Training details. We train TrustMark for 150 epochs with AdamW optimizer at an initial learning rate of $4e-6$ per image in a batch of 32s and cosine annealing schedule. Loss terms $(\beta_{\text{LPIPS}}, \beta_{\text{YUV}}, \beta_{\text{FFL}}, \beta_{\text{GAN}})$ are set to $(1, 1.5, 1.5, 1)$ – we do not turn them extensively. It takes 48 hours to train on a Geforce RTX 3090 GPU and a standard Intel i7 processor. For ReMark, we set n=3 and the training time is approx. 2 weeks for 100 epochs at batch size of 8 on an A100 GPU. For inference, average watermarking encoding/decoding takes 125/25 milliseconds on a Nvidia RTX 3090 GPU.

Since the watermark signal is small as opposed to the image content, it is important to prioritize the watermark extractor X at the early training phase. We set the trade-off parameter α low initially ($\alpha = 0.05$) and disable noise simulation and GAN loss as well as fixing the input image batch while varying random watermarks until X's detection accuracy exceeds a certain threshold. We then unlock subsequent TrustMark features (orders are: varying input batchs, enabling noise simulation and activating GAN module) before increasing α to the intended value α_{max} (more details in Sup.Mat.).

Metrics. We use standard PSNR for evaluating the imperceptibility of our watermarking/removal algorithms; and bit accuracy for watermark recovery on watermarked images after exposing to random noise (50% for random guess). Unless otherwise stated, we apply Algorithm 1 with default $\lambda = 1.0$ for the proposed methods and all baselines and compute these metrics at the original image resolution. We do not report other imperceptibility metrics such as SSIM or SFID [13] or bit accuracy on clean watermarked images since they are highly saturated for all methods.

Baselines. We compare TrustMark with recent watermark and steganography baselines including RoSteALS [13], RivaGAN [73], SSL [25], StegaStamp [66] and a traditional representative dwtDctSvd [51] (also used in [13]). We also report other baselines in Sup.Mat. For fair comparison, we retrain the baselines using the same noise simulation settings as TrustMark (Sec. 3.1) *if* it helps to improve performance. We also report TrustMark performance on their reported settings. At inference time, Resolution Scaling is applied to all methods except RivaGAN and the shallow methods that work at native resolution.

4.2. Watermark embedding

Tab. 1 shows performance of TrustMark (at $\alpha_{\text{max}} = 27.5$) and other baselines on the three benchmarks. Overall, TrustMark outperforms all baselines at every metrics on all benchmarks except PSNR on DIV2K (42.39dB) on par with SSL [25] (42.73dB). On the other hand, SSL robustness is in contrast with its imperceptibility performance, having the lowest bit accuracy among the deep learning approaches. RoSteALS [13] is the runner up in robustness but performs poorly in PSNR. RivaGAN [73] has the most balanced PSNR and bit accuracy scores among the baselines, yet underperforms TrustMark by a large margin. The shallow method dwtDctSvd [51] has near random bit accuracy due to the amount of noises involved, despite scoring reasonable PSNR. We note that it is not possible to control the imperceptibility-robustness trade-off for such hand-craft methods. Benchmark-wise, DIV2K proves to be the most challenging dataset for watermarking, while the narrow-domain MetFace yields the highest performance for most methods. Fig. 2 shows watermarking results for several cover images. TrustMark's artifacts are more uniform across color channels (RGB residual resembles a gray image), more invariant to semantic objects (it is harder to recognize image objects from the residual) and is overall less visible than other methods. We analyzed the mean loss of high frequency detail within 20% of the Nyquist limit, for DIV2K. TrustMark drops only 0.01% of the high frequencies vs. 0.05%: RoSteals; 0.07%: Riva-GAN; 0.93%: dwtDctSvd, preserving fine high-res details.

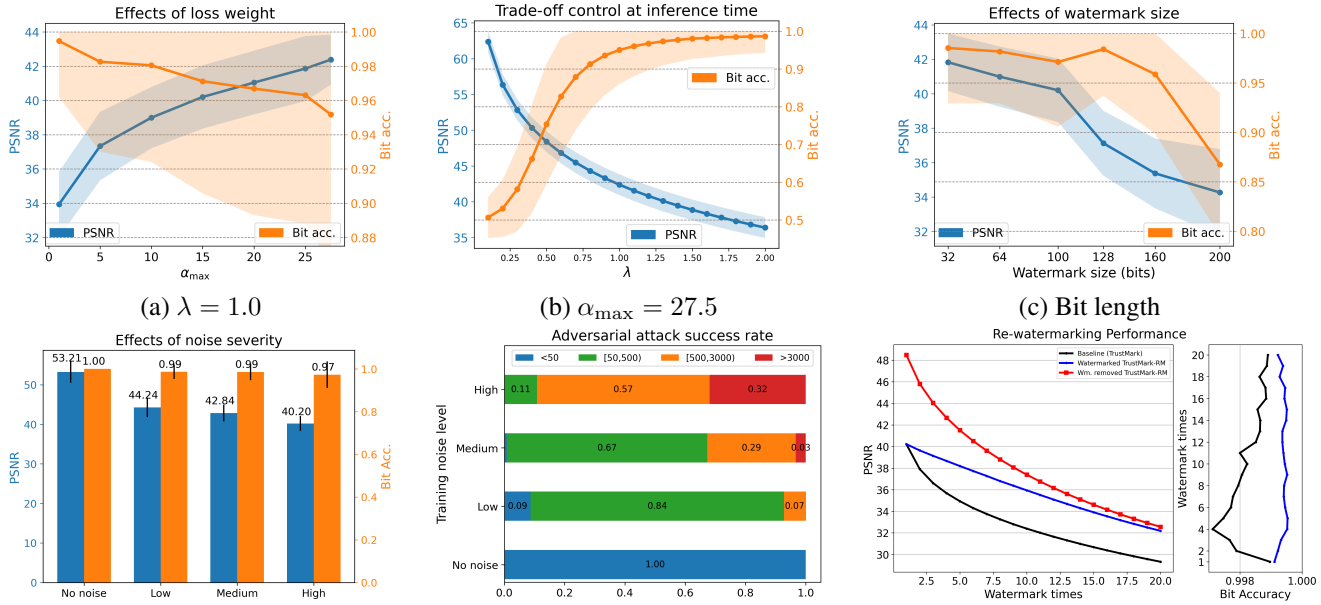
We also train and evaluate TrustMark using the same benchmark settings as the recent work of RoSteALS [13]. This benchmark is designed for robustness evaluation, using CLIC [67] dataset for cover images and destructive ImageNet-C transformations [33] commonly employed for classification evaluation as the noise sources. Resolution Scaling (Algorithm 1) is turned off to be compatible with the benchmark settings, meaning all performance metrics are computed at the model-designed resolution. We set $\alpha_{\text{max}} = 15$ for TrustMark to comfortably outperform RoSteALS on bit accuracy for this benchmark (Tab. 2), while achieving much higher PSNR (+6dB). Although RivaGAN and SSL have better imperceptibility scores, their robustness performance is significantly inferior to our method.

We highlight some reported baseline numbers by some recent localized region watermarking approaches; WAM (32-bit payload) [61] and OmniGuard (100-bit payload) [74]. These papers evaluate over distinct subsets of COCO reporting PSNR of 38.3dB and 42.3dB respectively, versus TrustMark PSNR of 40.3dB and 43.2dB on those subsets.

We study TrustMark's controllability over the imperceptibility-robustness trade-off in Fig. 3(a-b). At training time, the trade-off is controlled via the loss weight parameter α_{max} (Eq. (1)). We note the optimal range of α_{max} is (0,30) for stable training, as shown in Fig. 3(a). As α_{max} increases to 30, the PSNR improves by more than 8dB while bit accuracy drops by less than 5%. At test time, the trade-off can be controlled via the residual scale factor λ in Algorithm 1. Fig. 3(b) depicts the PSNR and bit accuracy trends over the optimal range of $\lambda = [0.1, 2.0]$, with bit accuracy increasing from random to near saturation at the cost of reducing PSNR from 63dB to 37dB. This controllability provides flexibility for end-users

Method	CLIC		DIV2K		MetFace	
	PSNR	Acc.	PSNR	Acc.	PSNR	Acc.
TrustMark ($\alpha_{\max} = 27.5$)	**43.26±1.59**	**0.95±0.09**	42.39±1.46	**0.95±0.09**	**45.34±1.33**	**0.96±0.10**
RoSteALS [13]	30.03±2.63	0.94±0.09	27.95±2.51	0.93±0.09	33.77±2.37	0.93±0.10
RivaGAN [73]	41.04±0.31	0.79±0.14	41.06±0.35	0.78±0.14	40.98±0.19	0.82±0.14
SSL [25]	42.74±0.12	0.60±0.09	**42.73±0.12**	0.57±0.07	42.84±0.10	0.70±0.13
StegaStamp [66]	37.48±1.93	0.72±0.10	35.87±1.73	0.70±0.10	39.35±1.57	0.72±0.11
dwtDctSvd [51]	39.13±1.21	0.52±0.06	38.02±1.35	0.51±0.06	41.14±2.35	0.52±0.08

Table 1. TrustMark versus baselines on three benchmarks. Performance metrics are PSNR for imperceptibility and bit accuracy for noised watermarked images for robustness. Best and runner up methods are marked with **bold** and underline.



(a) $\lambda = 1.0$ (b) $\alpha_{\max} = 27.5$ (c) Bit length

(d) Noise severity (e) Adversarial attack (f) Re-watermarking

Figure 3. Trade-off between imperceptibility and watermark recovery can be controlled at training (a) or deployment phases (b). Impact of bit length (c), noise severity (d) and adversarial attack (e) on performance. Re-watermarking with and without watermark removal (f).

Method	PSNR	Acc. (noised)
TrustMark ($\alpha_{\max} = 15$)	38.87±1.42	0.95±0.08
RoSteALS [13]	32.68 ± 1.75	0.94 ± 0.07
StegaStamp [66]	31.26 ± 0.85	0.88 ± 0.13
SSL [25]	41.84 ± 0.10	0.62 ± 0.14
RivaGAN [73]	40.32 ± 0.15	0.77 ± 0.16
dwtDctSvd [51]	38.96 ± 1.41	0.61 ± 0.20

Table 2. TrustMark versus baselines on CLIC dataset, using ImageNet-C noise configuration in training and evaluation. Baseline results are taken directly from [13].

to achieve desired watermark quality and robustness at individual image level.

4.3. Watermark length and robustness

Fig. 3 (c) shows TrustMark performance for the bit length range of [32, 200]. We fix $\alpha_{\max} = 20, \lambda = 1.0$ for this experiment. Overall, it is more challenging to embed and decode larger watermarks, as PSNR and bit accuracy both drop by 7.5dB and 11% when bit length increases 6 folds from 32 to 200, respectively.

We assess TrustMark robustness on several facets – against various noise sources, severity levels and adversarial attack. Fig. 4 shows bit accuracy of TrustMark and other baselines against every individual noise sources in Sec. 3.1. TrustMark outperforms the closest baseline RoSteALS on all sources except Gaussian noise and box blur. Other methods are robust against certain noises but weak against others *e.g.* dwtDctSvd performs well for Gaussian blur and Posterize but close to random chance for Grayscale or RGB shift.

To evaluate noise severity, we train and test TrustMark on 3 additional variants of noise settings: no noise simulation, low-level noise and medium-level noise as demonstrated in Fig. 3(d). Increasing noise severity affects PSNR the most while bit accuracy stays roughly the same. Specifically, PSNR sets at 53.2dB without noise simulation then drops to 40.2dB for high severity noise, but bit accuracy

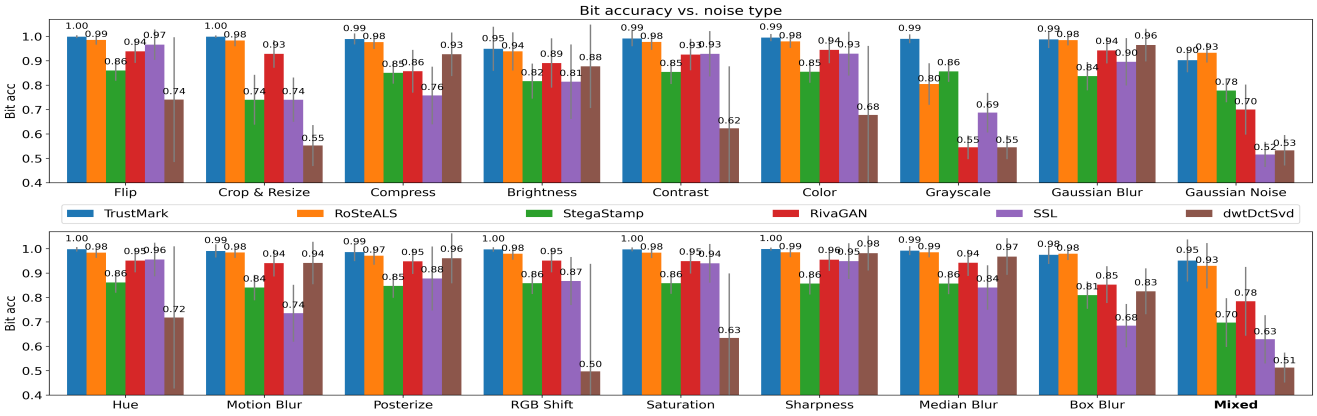


Figure 4. Characterizing the robustness of TrustMark and baselines to individual noise perturbations.

Method	PSNR↑	Bit Acc. (~ 0.5)
No removal	40.20	0.970
ReMark	48.48	0.553
Pinto *et al*. [57]	40.75	0.550
Hönig *et al*. [36]	39.93	0.830
I-FGSM [30]	23.48	0.629
Zhao *et al*. [76] ($\sigma = 0.3$)	9.23	0.660

Table 3. Watermark removal for TrustMark ($\alpha_{\max} = 20$).

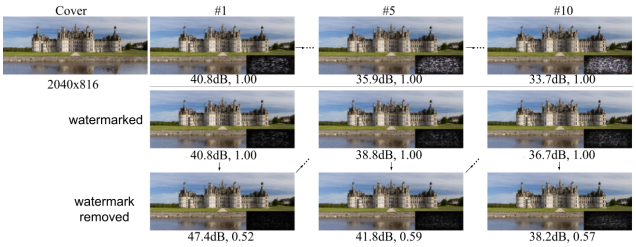


Figure 5. Left to right: Re-watermarking a cover image 1,5 and 10 times without (first row) and with (2nd and 3rd row) watermark removal. Inset: 20× residual. Please zoom to inspect details.

reduces by only 3%. We attribute this behavior to the end-to-end training of TrustMark embedder and extractor, where the gradient of the recovery loss $\mathcal{L}_{\text{recovery}}(.)$ in Eq. (1) affects both modules at the same time.

Using high severity noise simulation during training also makes TrustMark more robust to adversarial attack, as shown in Fig. 3(e). Here, we perform I-FGSM attack [30] by adding subtle noise with maximum strength $\epsilon = 8/255$ into the watermarked image to fool the watermark extraction model. The adversarial noise is initially set to 0 then is adjusted at each attack iteration until the bit accuracy of the target image is brought down below $0.5 + \epsilon/2$, regardless of PSNR. The number of I-FGSM iterations reflects the robustness of the watermark model against adversarial attack. Per Fig. 3(e), when TrustMark is trained without noise simulation, it takes less than 50 iterations for a successful attack on any watermarked image. In contrast, when trained with high level noises, 32% of the watermarked images require more than 3000 attack iterations.

4.4. Watermark removal and re-watermarking

Watermark removal We compare ReMark with adversarial attack baselines, Pinto *et al*. [57], Hönig *et al*. [36], I-FGSM [30], and the denoising-based watermark removal work by Zhao *et al*. [76]. The target watermarking model for all removal methods is TrustMark with $\alpha_{\max} = 20, \lambda = 1.0$. Tab. 3 shows that ReMark not only brings down bit accuracy to 55% but also improves PSNR to 48.5dB. In contrast, Pinto *et al*. and I-FGSM only succeed in bit accuracy because their sole aim is to 'break' the decoder module. Hönig *et al*. [36] only partially removes the signal. Similarly weaker performance (and also low PSNR) of Zhao *et al*. [76] demonstrate the resilience of TrustMark to arbitrary image-denoising attacks and the need for ReMark for high quality re-watermarking. ReMark is selective at targetted removal of the TrustMark signal whilst retaining other detail (i.e. high PSNR) including methods it is not trained on.

Re-watermarking Fig. 3(f) demonstrates ReMark efficacy for re-watermarking. We make 2 observations: (i) bit accuracy is not affected if a watermark remover is employed or not (Fig. 3(f) right); and (ii) ReMark preserves image quality better than not using it (Fig. 3(f) left). However, the denoising effect of ReMark is weakened after each re-watermarking, because the unwanted noise generated by ReMark gets accumulated over time. A re-watermarking example is illustrated in Fig. 5.

4.5. Ablation study

Tab. 4 shows the contribution of each design components. Our training strategy ensures watermark recovery is always prioritized in the first training phase, enabling bit accuracy performance to be maintained through various ablations. The architecture mutations influence PSNR mostly. When GAN, $\mathbf{E}_{\text{post}}$ and FFL loss are all disabled, TrustMark PSNR is equivalent to StegaStamp [66] (exp (i)). Adding GAN,

	(i)	(ii)	(iii)	(iv)	(v)	(vi)	(vii)	(viii)	[59]	[71]	[32]	[37]
GAN			✓		✓	✓		✓	✓	✓	✓	✓
FFL		✓			✓		✓	✓	✓	✓	✓	✓
$\mathbf{E}_{\text{post}}$				✓		✓	✓	✓	✓	✓	✓	✓
PSNR	36.30	38.08	37.03	38.92	38.37	39.53	39.60	**40.20**	39.26	39.46	38.86	39.57
Acc.	0.959	0.974	0.963	0.979	0.970	0.974	0.975	0.973	**0.986**	0.976	0.972	0.982

Table 4. Ablation studies on different mutations of TrustMark ($\alpha_{\max} = 20, \lambda = 1.0$) architecture. Ablated backbones are RegNet [59], ResNext [71], ResNet18 [32], DenseNet121 [37].

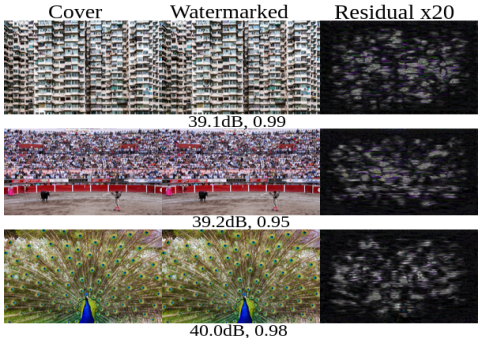


Figure 6. Limitations. TrustMark can watermark any image. Occasionally very cluttered images reduce bit accuracy by a couple of %, mitigated by raising the strength of the residual.

$\mathbf{E}_{\text{post}}$ and FFL separately improves the score by 0.7dB, 1.6dB and 1.7dB respectively (exp (ii-iv)). $\mathbf{E}_{\text{post}}$ combined with either GAN or FFL boosts PSNR by 3dB (exp (v-vii)) and all three components make up 4dB in PSNR and 1.4% in bit accuracy in total (exp (viii)).

We also experiment with different backbones for our watermark decoder $\mathbf{X}$. TrustMark training converges for ResNet family, including ResNet18 [32], DenseNet121 [37], RegNet [59] and ResNext [71], but is not successful for VGG [62], GoogleNet [23], ConvNext [46] and EfficientNet [65]. We observe all successful backbones have either a residual layer or a skip layer - both allow signals from bottom layers to flow directly to the top via a sum (residual) or concatenation (skip) operation. We attribute this unique requirement of TrustMark to the complexity of our multi-noise simulation scheme and the accuracy-thresholded multi-stage training procedure.

Finally, we evaluated high and arbitrary resolution watermarking by ablating the DIV2K dataset to 20% to 100% of original (2K) resolution. We observe after encoding that PSNR varies only by ± 0.02dB and after decoding that bit accuracy varies only by $\pm 10^{-4}$ on average across all resolutions. TrustMark shows near-equivalent performance across arbitrary resolutions due to our Resolution Scaling technique (Sec. 3.3). In Sup. Mat. we show that disabling scaling maintains bit accuracy but significantly lowers PSNR.

Practical Uses and Limitations. We have found TrustMark to perform consistently well down at image resolution of all sizes down to 80px shortest side, which presents a practical lower limit. We examined the encoded images with lowest PSNR scores and observe that all are highly cluttered (Fig. 6). TrustMark already alleviates this thanks to our FFL loss (>39dB PSNR at worst). The non-perfect bit accuracy can be improved by: 1) an error correcting code such as BCH [7] (not used in any results reported, but available in our open source code [11]); 2) raising the control factor λ in Algorithm 1 to strengthen the watermark. As for ReMark, we make a positive use case for re-watermarking but acknowledge that it could be used to spoof identifiers used to recover provenance metadata. Known mitigations include incorporating a visual check between the watermarked image and a fingerprint [19] or thumbnail within recovered metadata, a workflow first proposed by Adobe [17] who are one of several commercial adopters now using TrustMark for C2PA provenance lookup (see video supmat.). As a robust, high visual quality solution to image watermarking for provenance, TrustMark's open source implementation [11] is becoming widely adopted since the preprint release of this paper on arXiv [12] (Nov, 2023) and is included on the C2PA 'soft binding algorithm list' [16].

5. Conclusion

We propose TrustMark and ReMark for watermarking and watermark removal. TrustMark integrates novel designs in architecture and losses and rigorous noise simulation for robustness. With ReMark, we show that a customized denoising network is needed to restore a high quality image from the TrustMark watermarked image. We propose an effective resolution scaling algorithm to extend TrustMark and ReMark for images with arbitrary resolution and enables trade-off control at inference time. We show TrustMark encoded images are imperceptible (PSNR$>$ 40dB) while being state-of-art robust to various noise sources and can be restored with high fidelity (PSNR$>$ 48dB). Applications of this work include content authenticity workflows where identifiers may be imperceptibly embedded to track the provenance of image assets in conjunction with open standards such as C2PA [16]. Future work could include other uses such as training data encoding to tackle GenAI model attribution [2, 10] or extensions to video via further noise augmentation similar to [27]. TrustMark has also been shown to co-exist well with other watermarks [55], and may aid watermark interoperability [18].

Acknowledgment This work was supported by DECaDE under UKRI/EPSRC grant EP/T022485/1.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proc. CVPR Workshops*, 2017. 4, 5

[2] Vishal Asnani, John Collomosse, Tu Bui, Xiaoming Liu, and Shruti Agarwal. Promark: Proactive diffusion watermarking for causal attribution. In *Proc. CVPR*, 2024. 2, 8

[3] J. Aythora et al. Multi-stakeholder media provenance management to counter synthetic media risks in news publishing. In *Proc. Intl. Broadcasting Convention (IBC)*, 2020. 1

[4] Shumeet Baluja. Hiding images in plain sight: Deep steganography. *NeurIPS*, 30, 2017. 3

[5] A. Bharati, D. Moreira, P. Flynn, A. de Rezende Rocha, K. Bowyer, and W. Scheirer. Transformation-aware embeddings for image provenance. *IEEE Trans. Info. Forensics and Sec.*, 16:2493–2507, 2021. 1

[6] A. Black, T. Bui, H. Jin, V. Swaminathan, and J. Collomosse. Deep image comparator: Learning to visualize editorial change. In *Proc. CVPR WMF*, pages 972–980, 2021. 1

[7] Raj Chandra Bose and Dwijendra K Ray-Chaudhuri. On a class of error correcting binary group codes. *Information and control*, 3(1):68–79, 1960. 8

[8] T. Bui, D. Cooper, J. Collomosse, M. Bell, A. Green, J. Sheridan, J. Higgins, A. Das, J. Keller, O. Thereaux, and A. Brown. Archangel: Tamper-proofing video archives using temporal content hashes on the blockchain. In *Proc. CVPR Workshop CV, AI and Blockchain*, 2019. 1

[9] T. Bui, D. Cooper, J. Collomosse, M. Bell, A. Green, J. Sheridan, J. Higgins, A. Das, J. Keller, and O. Thereaux. Tamper-proofing video with hierarchical attention autoencoder hashing on blockchain. *IEEE Trans. Multimedia (TMM)*, 22(11):2858–2872, 2020. 1

[10] Tu Bui, Ning Yu, and John Collomosse. Repmix: Representation mixing for robust attribution of synthesized images. In *ECCV*, pages 146–163. Springer, 2022. 4, 8

[11] T. Bui, S. Agarwal, and J. Collomosse. Trustmark github open-source code release (MIT license). https://github.com/adobe/trustmark, 2023. 8

[12] Tu Bui, Shruti Agarwal, and John Collomosse. Trustmark: Universal watermarking for arbitrary resolution images. *ArXiv e-prints*, 2023. 8

[13] Tu Bui, Shruti Agarwal, Ning Yu, and John Collomosse. Rosteals: Robust steganography using autoencoder latent space. In *Proc. CVPR WMF*, pages 933–942, 2023. 1, 2, 3, 4, 5, 6

[14] Ching-Chun Chang. Neural reversible steganography with long short-term memory. *Security and Communication Networks*, 2021:1–14, 2021. 2

[15] Jianbo Chen, Xinwei Liu, Siyuan Liang, Xiaojun Jia, and Yuan Xun. Universal watermark vaccine: Universal adversarial perturbations for watermark protection. In *Proc. CVPR*, pages 2321–2328, 2023. 2

[16] Coalition for Content Provenance and Authenticity. Technical specification 1.3. Technical report, C2PA, 2023. 1, 8

[17] John Collomosse. The three pillars of provenance that make up durable content credentials. https://contentauthenticity.org/blog/three-pillars-of-provenance, 2024. 8

[18] John Collomosse and Dom Guinard. Digital watermarking for interoperable and durable content credentials. https://contentauthenticity.org/blog/digital-watermarking-interoperable-durable-content-credentials, 2025. 8

[19] J. Collomosse and A. Parsons. To authenticity, and beyond! building safe and fair generative ai upon the three pillars of provenance. *IEEE Comp. Graphics and Appl.*, 44(3):82–90, 2024. 1, 8

[20] Xiaodong Cun and Chi-Man Pun. Split then refine: stacked attention-guided resunets for blind single image visible watermark removal. In *Proc. AAAI*, pages 1184–1192, 2021. 2

[21] Xintao Duan, Kai Jia, Baoxia Li, Daidou Guo, En Zhang, and Chuan Qin. Reversible image steganography scheme based on a u-net structure. *IEEE Access*, 7:9314–9323, 2019. 2

[22] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proc. CVPR*, pages 12873–12883, 2021. 2

[23] Szegedy etal. Going deeper with convolutions. In *Proc. CVPR*, pages 1–9, 2015. 8

[24] Jianwei Fei, Zhihua Xia, Benedetta Tondi, and Mauro Barni. Supervised GAN watermarking for intellectual property protection. In *Proc. WIFS*. IEEE, 2022. 2

[25] Pierre Fernandez, Alexandre Sablayrolles, Teddy Furon, Hervé Jégou, and Matthijs Douze. Watermarking images in self-supervised latent spaces. In *Proc. ICASSP*, pages 3054–3058. IEEE, 2022. 2, 4, 5, 6

[26] Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *Proc. ICCV*. IEEE, 2023. 2

[27] Pierre Fernandez, Hady Elsahar, I. Zeki Yalniz, and Alexandre Mourachko. Video seal: Open and efficient video watermarking. *ArXiv e-prints*, 2024. 8

[28] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *Proc. ICML*, pages 3247–3258. PMLR, 2020. 4

[29] Kazem Ghazanfari, Shahrokh Ghaemmaghami, and Saeed R Khosravi. Lsb++: An improvement to lsb+ steganography. In *TENCON 2011-2011 IEEE Region 10 Conference*, pages 364–368. IEEE, 2011. 2

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

[31] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *NeurIPS*, 30, 2017. 4

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, pages 770–778, 2016. 3, 8

[33] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *Proc. ICLR*, 2019. 5

[34] Vojtěch Holub and Jessica Fridrich. Designing steganographic distortion using directional filters. In *WIFS*, pages 234–239. IEEE, 2012. 2

[35] Vojtěch Holub, Jessica Fridrich, and Tomáš Denemark. Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security*, 2014: 1–13, 2014. 2

[36] R. Hönig, J. Rando, N. Carlini, and F. Tramèr. Adversarial perturbations cannot reliably protect artists from generative AI. In *Proc. ICLR*, 2025. 7

[37] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proc. CVPR*, pages 4700–4708, 2017. 8

[38] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proc. ECCV*, pages 172–189, 2018. 3

[39] Mark J Huiskes and Michael S Lew. The mir flickr retrieval evaluation. In *Proc. ICMIR*, pages 39–43, 2008. 5

[40] Zhaoyang Jia, Han Fang, and Weiming Zhang. Mbrs: Enhancing robustness of dnn-based watermarking by mini-batch of real and simulated jpeg compression. In *Proc. ACM MM*, pages 41–49, 2021. 1, 4

[41] Liming Jiang, Bo Dai, Wayne Wu, and Chen Change Loy. Focal frequency loss for image reconstruction and synthesis. In *Proc. ICCV*, pages 13919–13929, 2021. 4

[42] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *NeurIPS*, 33:12104–12114, 2020. 4, 5

[43] Xiaoxia Li and Jianjun Wang. A steganographic method based upon jpeg and particle swarm optimization algorithm. *Information Sciences*, 177(15):3099–3109, 2007. 2

[44] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013. 3

[45] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proc. ICCV*, pages 2980–2988, 2017. 4

[46] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proc. CVPR*, pages 11976–11986, 2022. 8

[47] C.-S. Lu, S.-W. Sun, C.-Y. Hsu, and P.-C. Chang. Media hash-dependent image watermarking resilient against both geometric attacks and estimation attacks based on false positive-oriented detection. *IEEE Trans. Multimedia*, 8(4), 2006. 2

[48] Xiyang Luo, Ruohan Zhan, Huiwen Chang, Feng Yang, and Peyman Milanfar. Distortion agnostic deep watermarking. In *Proc. CVPR*, pages 13548–13557, 2020. 1

[49] Mingzhi Lyu, Yi Huang, and Adams Wai-Kin Kong. Adversarial attack for robust watermark protection against inpainting-based and blind watermark removers. In *Proc. Int. Conf. Multimedia*, pages 8396–8405, 2023. 2

[50] Ruohan Meng, Steven G Rice, Jin Wang, and Xingming Sun. A fusion steganographic algorithm based on faster r-cnn. *Computers, Materials & Continua*, 55(1):1–16, 2018. 2

[51] KA Navas, Mathews Cheriyan Ajay, M Lekshmi, Tampy S Archana, and M Sasikumar. Dwt-dct-svd based watermarking. In *COMSWARE'08*, pages 271–274. IEEE, 2008. 2, 3, 4, 5, 6

[52] E. Nguyen, T. Bui, V. Swaminathan, and J. Collomosse. Oscar-net: Object-centric scene graph attention for image attribution. In *Proc. ICCV*, 2021. 1

[53] Li Niu, Xing Zhao, Bo Zhang, and Liqing Zhang. Fine-grained visible watermark removal. In *Proc. ICCV*, pages 12770–12779, 2023. 2

[54] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn. Attacks on copyright marking systems. In *Proc. Intl. Workshop on Information Hiding (IH)*, pages 219–239, 1998. 2

[55] Aleksandar Petrov, Shruti Agarwal, Philip Torr, Adel Bibi, and John Collomosse. On the coexistence and ensembling of watermarks. In *Proc. ICLR Workshop on Watermarking (ICLR WMARK)*, 2025. 8

[56] Tomáš Pevnỳ, Tomáš Filler, and Patrick Bas. Using high-dimensional image models to perform highly undetectable steganography. In *Proc. Int. Conf. Information Hiding*, pages 161–177. Springer, 2010. 2

[57] Maura Pintor, Fabio Roli, Wieland Brendel, and Battista Biggio. Fast minimum-norm adversarial attacks through adaptive norm constraints. *NeurIPS*, 34:20052–20062, 2021. 7

[58] Niels Provos. Defending against statistical steganalysis. In *Usenix security symposium*, pages 323–336, 2001. 2

[59] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proc. CVPR*, pages 10428–10436, 2020. 8

[60] L. Rosenthol, A. Parsons, E. Scouten, J. Aythora, B. MacCormack, P. England, M. Levallee, J. Dotan, et al. Content authenticity initiative (CAI): Setting the standard for content attribution. Technical report, Adobe Inc., 2020. 1

[61] Tom Sander, Pierre Fernandez, Alain Oliviero Durmus, Teddy Furon, and Matthijs Douze. Watermark anything with localized messages. In *Proc. ICLR*, 2025. 2, 5

[62] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 8

[63] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. CVPR*, pages 1–9, 2015. 3

[64] Mustafa Sabah Taha, Mohd Shafry Mohd Rahem, Mohammed Mahdi Hashim, and Hiyam N Khalid. High payload image steganography scheme with minimum distortion based on distinction grade value method. *Multimedia Tools and Applications*, 81(18):25913–25946, 2022. 2

[65] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proc. ICML*, pages 6105–6114. PMLR, 2019. 8

[66] Matthew Tancik, Ben Mildenhall, and Ren Ng. Stegastamp: Invisible hyperlinks in physical photographs. In *Proc. CVPR*, pages 2117–2126, 2020. 1, 2, 3, 4, 5, 6, 7

[67] G Toderici, W Shi, R Timofte, L Theis, J Ballé, E Agustsson, Nick Johnston, and F Mentzer. Workshop and challenge on learned image compression (clic2020). In *CVPR*, 2020. 4, 5

[68] Wenbo Wan, Jun Wang, Yunming Zhang, Jing Li, Hui Yu, and Jiande Sun. A comprehensive survey on robust image watermarking. *Neurocomputing*, 2022. 2

[69] Raymond B Wolfgang and Edward J Delp. A watermark for digital images. In *Proc. ICIP*, pages 219–222. IEEE, 1996. 2

[70] Pin Wu, Yang Yang, and Xiaoqiang Li. Stegnet: Mega image steganography capacity with deep convolutional network. *Future Internet*, 10(6):54, 2018. 2

[71] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proc. CVPR*, pages 1492–1500, 2017. 8

[72] Ning Yu, Vladislav Skripniuk, Sahar Abdelnabi, and Mario Fritz. Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In *Proceedings of the IEEE/CVF International conference on computer vision*, pages 14448–14457, 2021. 2

[73] Kevin Alex Zhang, Lei Xu, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Robust invisible video watermarking with attention. *arXiv preprint arXiv:1909.01285*, 2019. 1, 2, 3, 4, 5, 6

[74] Xuanyu Zhang, Zecheng Tang, Zhipei Xu, Runyi Li, Youmin Xu, Bin Chen, Feng Gao, and Jian Zhang. Omniguard: Hybrid manipulation localization via augmented versatile deep image watermarking. In *Proc. CVPR*, 2025. 2, 5

[75] Yi Zhang, Dasong Li, Xiaoyu Shi, Dailan He, Kangning Song, Xiaogang Wang, Hongwei Qin, and Hongsheng Li. Kbnet: Kernel basis network for image restoration. *arXiv preprint arXiv:2303.02881*, 2023. 4

[76] Xuandong Zhao, Kexun Zhang, Zihao Su, Saastha Vasan, Ilya Grishchenko, Christopher Kruegel, Giovanni Vigna, Yu-Xiang Wang, and Lei Li. Invisible image watermarks are provably removable using generative AI, 2023. 2, 4, 7

[77] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *Proc. ECCV*, pages 657–672, 2018. 1, 2, 3, 4