Watermark Aging of Stable Signature in Latent Diffusion Models

Shreyas Prasad

Introduction

- Stable Diffusion is superior in image synthesis
- The proliferation of Generative AI content poses problems
  - Copyright protection
  - Ownership
  - Transparency
- Replication and Distribution of multimedia content

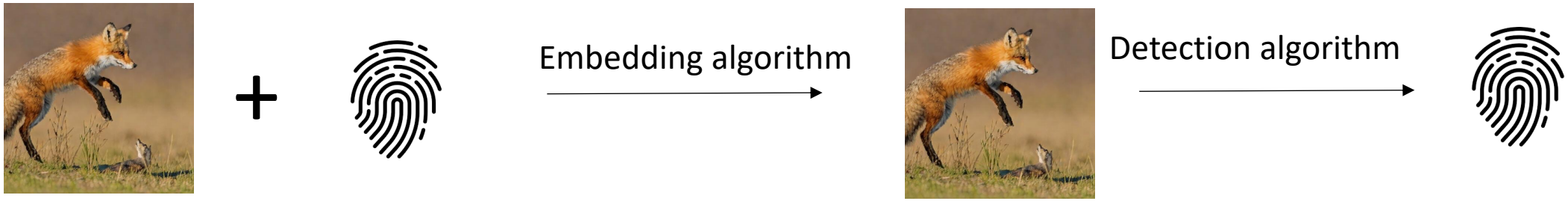
Verification of authenticity and originality is important

Watermarking

- A signature covertly embedded in a noise-tolerant signal (e.g. Images)
- To identify the creator and protect the image from unauthorized use.

Why watermarking images after generation methods do not work?

- Model leaks
- Open source



Related Work

Watermarking with Deep Networks

- *HiDDen: Hiding Data With Deep Network [Zhu et al.]*
- *Distortion Agnostic Deep Watermarking [Luo et al.]*

How it works?

- Jointly train encoder and decoder networks
- given an input message and cover image, decoder can recover the original message

Watermarking in generative models

- *Supervised GAN Watermarking for Intellectual Property Protection*
- *CycleGANWM: A CycleGAN watermarking method for ownership verification*

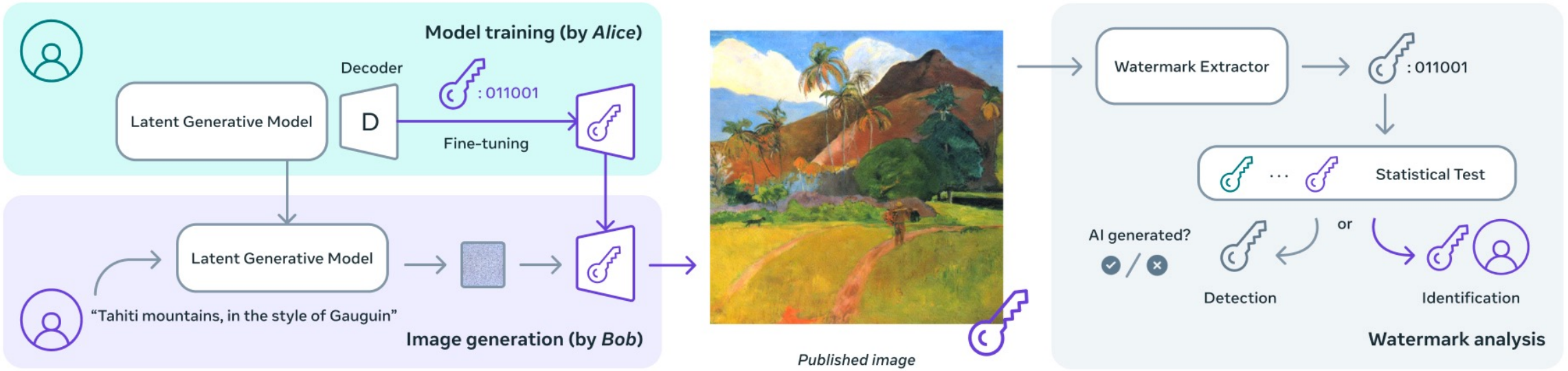
Challenges in Expanding to Stable Diffusion Models

- Stable Diffusion models are still under-explored.
- GAN methods are high cost and effort.

Stable Signature

How does it work?

- A watermark extractor network (HiDDen) is created and trained to embed and extract watermarks from images.
- The decoder of a Latent Diffusion Model is fine-tuned so that all images it generates contain a predetermined watermark



https://ai.meta.com/blog/stable-signature-watermarking-generative-ai/

Watermark Aging

- Media on the internet do not remain static
- Images under successive transformations, edits, and filters which may degrade over time
- For example, this would involve repeated compressions, the impact of different format transition
- Stable Signature does not evaluate its model against this phenomenon

Fig. An example of successive transformation of an image

Objectives

- Re-implement the Stable Diffusion pipeline
- Run evaluations and compare to the original pipeline
- Extend the evaluation to “aging” of watermarks (successive and combinations) to further add to the evaluations

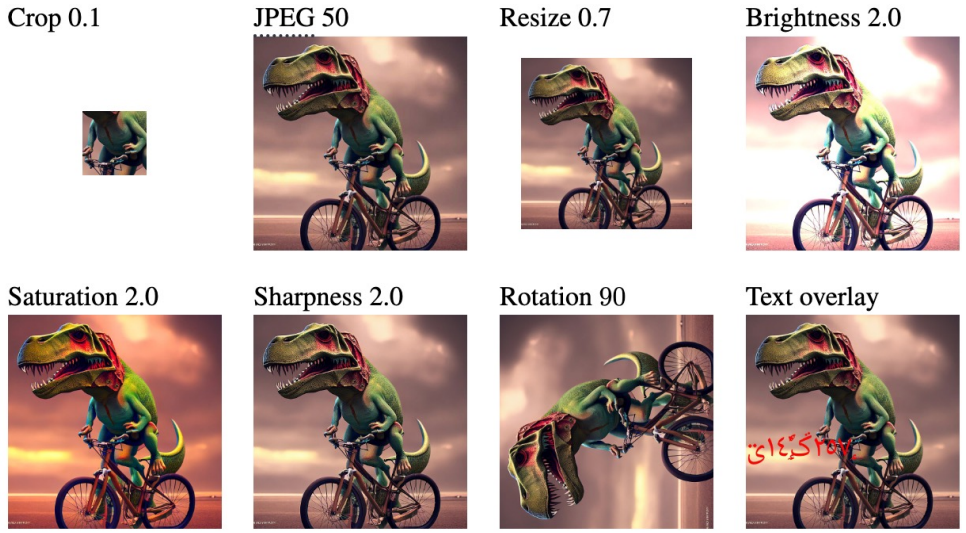


Figure 9. Illustration of all transformations evaluated

Methodology and Experimentation

1. **Pre-training the Watermark Extractor**
   - Encode-Decoder which embeds a 48-bit message in the image
     3x (Conv + Batch + GELU), 1000 images
   - Binary Cross Entropy is calculated on the 48-bit extracted message and the ground truth to optimize the embedding and extraction process

2. **Fine-tuning of LDM**
   - Use 1000 images from the previous step to fine-tune the decoder of an LDM (SD Turbo)
   - Loss functions:
     - Binary Cross Entropy for the message
     - Watson-VGG perceptual loss to control image distortion
   - Iterations: 100, AdamW, batch = 4, lr = 10e-4

3. **Evaluation and Detection of watermarking**
   - For each image, the extracted message '*m*' is compared with the known signature.
   - H_0: The null hypothesis image was not generated by the watermarked model.
   - Under H_0, assume that bits are (i.i.d.) Bernoulli random variables (0.5)
   - Calculate TPR and FPR under all thresholds, threshold being "no_of_bits"

New: Evaluation involves sequential transformations

Results

Crop 0.1

JPEG 50

Resize 0.7

Brightness 2.0

Saturation 2.0

Sharpness 2.0

Rotation 90

Text overlay

Meta's Original (Internal LDM):

Crop 0.1

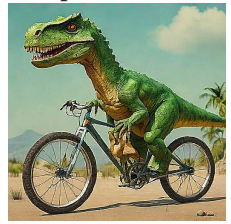
JPEG 50

Resize 0.7

Brightness 2.0

Saturation 2.0

Sharpness 2.0



Rotation 90

Text overlay

Our Implementation (SD Turbo):

Results

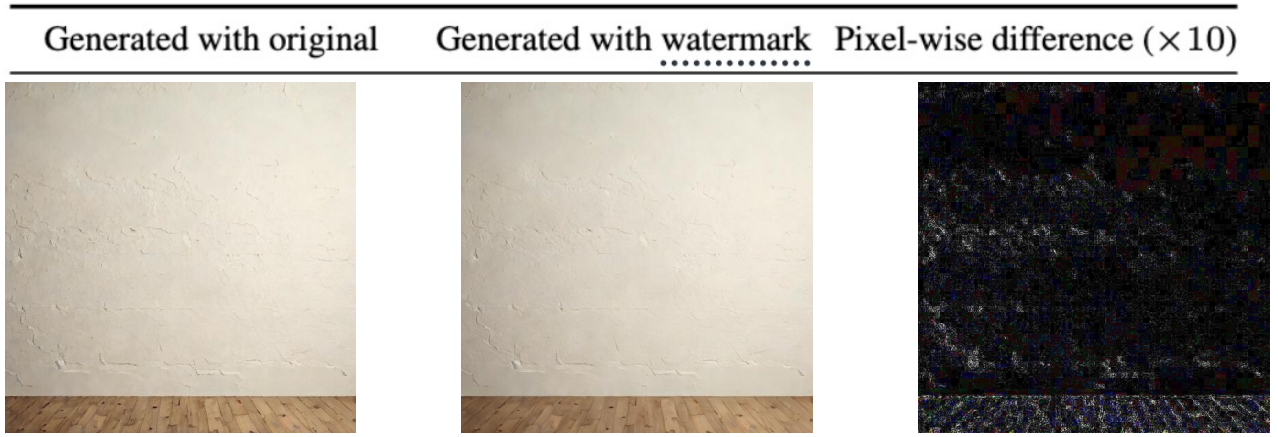


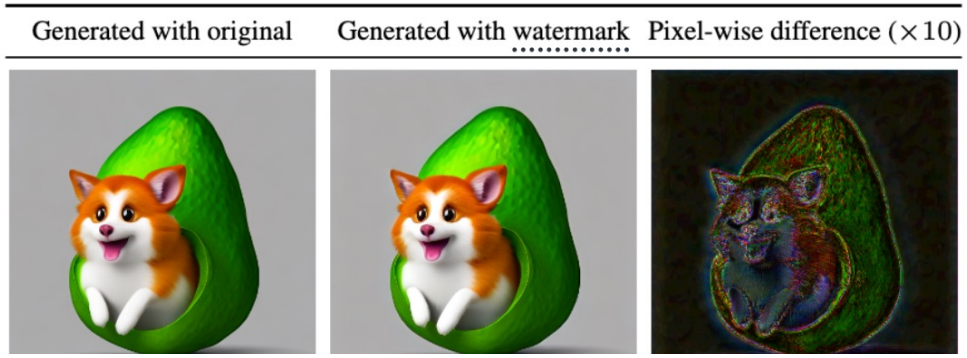
Fig. An example of an image when the model fails:
Not Detected

Results

Watermark robustness on different tasks and image transformations applied before decoding

- Original:
  - The bit accuracy averaged over 10x1k images generated with 10 different keys.
  - LDM trained by Meta is internal and not public.
- Our:
  - The bit accuracy averaged over 1x1k images generated with 1 key.

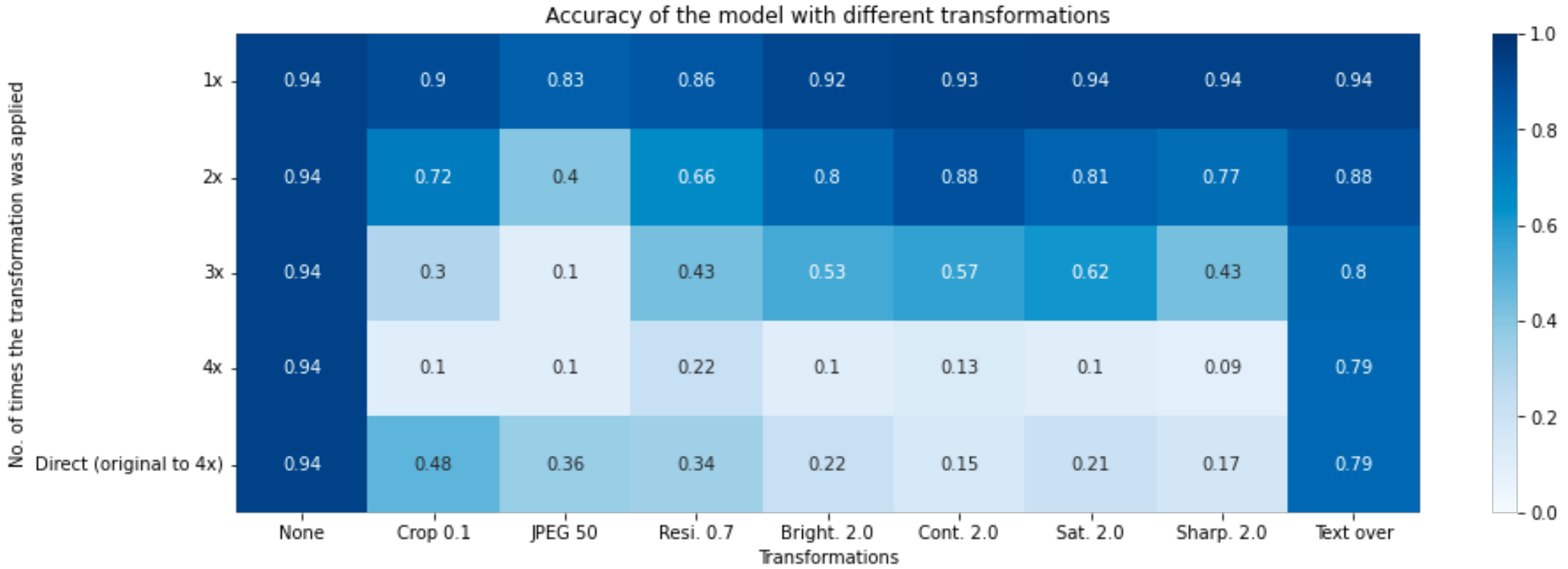
Meta's Original (Internal LDM):



Our Implementation (SD Turbo):

Task		None	Crop 0.1	JPEG 50	Resi. 0.7	Bright. 2.0	Cont. 2.0	Sat. 2.0	Sharp. 2.0	Text over.
Text-to-Image	LDM	0.99	0.95	0.88	0.91	0.97	0.98	0.99	0.99	0.99
Our Implementation	Stable Diffusion Turbo	0.94	0.90	0.83	0.86	0.92	0.93	0.94	0.94	0.94

Results


Accuracy of the model with different transformations
No. of times the transformation was applied
1x
2x
3x
4x
Direct (original to 4x)
0.94 0.9 0.83 0.86 0.92 0.93 0.94 0.94 0.94
0.94 0.72 0.4 0.66 0.8 0.88 0.81 0.77 0.88
0.94 0.3 0.1 0.43 0.53 0.57 0.62 0.43 0.8
0.94 0.1 0.1 0.22 0.1 0.13 0.1 0.09 0.79
0.94 0.48 0.36 0.34 0.22 0.15 0.21 0.17 0.79
None
Crop 0.1
JPEG 50
Resi. 0.7
Bright. 2.0
Cont. 2.0
Sat. 2.0
Sharp. 2.0
Text over
Transformations
1.0
0.8
0.6
0.4
0.2
0.0

Conclusion

Questions?