Trustworthy AI Systems

-- Accountability of AI

Instructor: Guangjing Wang

guangjingwang@usf.edu

Last Lecture

- Membership Inference Attacks
- Model Inversion Attacks
- Model Stealing Attacks
- Privacy Protection Methods

This Lecture

- Accountability
- Detecting AI-generated Content
- Watermarking Techniques
- Evading Watermarking-based Detection

Accountability of AI

If an AI system causes harm (e.g., a self-driving car accident), is it the company deploying the system, the developer or the user liable?

- Companies: Companies deploying AI systems must ensure they follow legal and ethical guidelines, and mitigate risks such as discrimination, misinformation, or unintended consequences.

- Developers: Responsible for ensuring that AI systems are fair, safe, transparent, and do not perpetuate harm or bias.

- Users: Individuals or entities that use AI systems responsibly. They should understand the limitations of AI, follow ethical guidelines in its application, and avoid misuse.

Key Questions about AI Accountability

- What is accountability?
  - An obligation or willingness to accept responsibility or to account for one's action.
- How to achieve accountability?
  - AI systems need to be transparent (people should understand how decisions are made) and explainable (the decision-making process should be clear enough).
  - Regulatory frameworks should be provided to ensure accountability.

Challenges in AI Accountability

- Complexity of AI Systems: Advanced AI systems (e.g., deep learning) often function as black boxes, making it difficult to trace specific decision-making steps.

- Shared Responsibility: AI systems often rely on multiple stakeholders (developers, users, etc.), so it can be hard to pinpoint exactly who is responsible when something goes wrong.

- Autonomous Decision-Making: Making decisions without human intervention challenges traditional notions of accountability, as the AI may make choices its creators did not explicitly intend.

This Lecture

- Accountability
- Detecting AI-generated Content
  - An important aspect of accountability of AI
- Watermarking Techniques
- Evading Watermarking-based Detection

AI-generated Content VS Deepfake

- AI-generated content refers to any content created by AI algorithms, e.g. **text** generation, **image** and art creation, **video** and animation, **music** composition, and **voice** synthesis. **(broad term)**

- Deepfakes specifically refer to synthetic media where **a person in an existing image or video** is replaced with someone else's likeness. It uses deep learning techniques to make these alterations appear as realistic as possible.

Generative AI for Image

Generative model	Size	Number	Source
ProGAN [19]	256×256	8.0k	LSUN
StyleGAN [20]	256×256	12.0k	LSUN
BigGAN [3]	256×256	4.0k	ImageNet
CycleGAN [52]	256×256	2.6k	ImageNet
StarGAN [6]	256×256	4.0k	CelebA
GauGAN [33]	256×256	10.0k	COCO
Stylegan2 [21]	256×256	15.9k	LSUN
WFIR [49]	1024×1024	2.0k	FFHQ
ADM [9]	256×256	12.0k	ImageNet
Glide [31]	256×256	12.0k	ImageNet
Midjourney [30]	1024×1024	12.0k	ImageNet
SDv1.4 [39]	512×512	12.0k	ImageNet
SDv1.5 [39]	512×512	16.0k	ImageNet
VQDM [15]	256×256	12.0k	ImageNet
Wukong [50]	512×512	12.0k	ImageNet
DALL-E 2 [37]	256×256	2k	ImageNet

https://www.whichfaceisreal.com/ (StyleGAN)

Images from StyleGAN

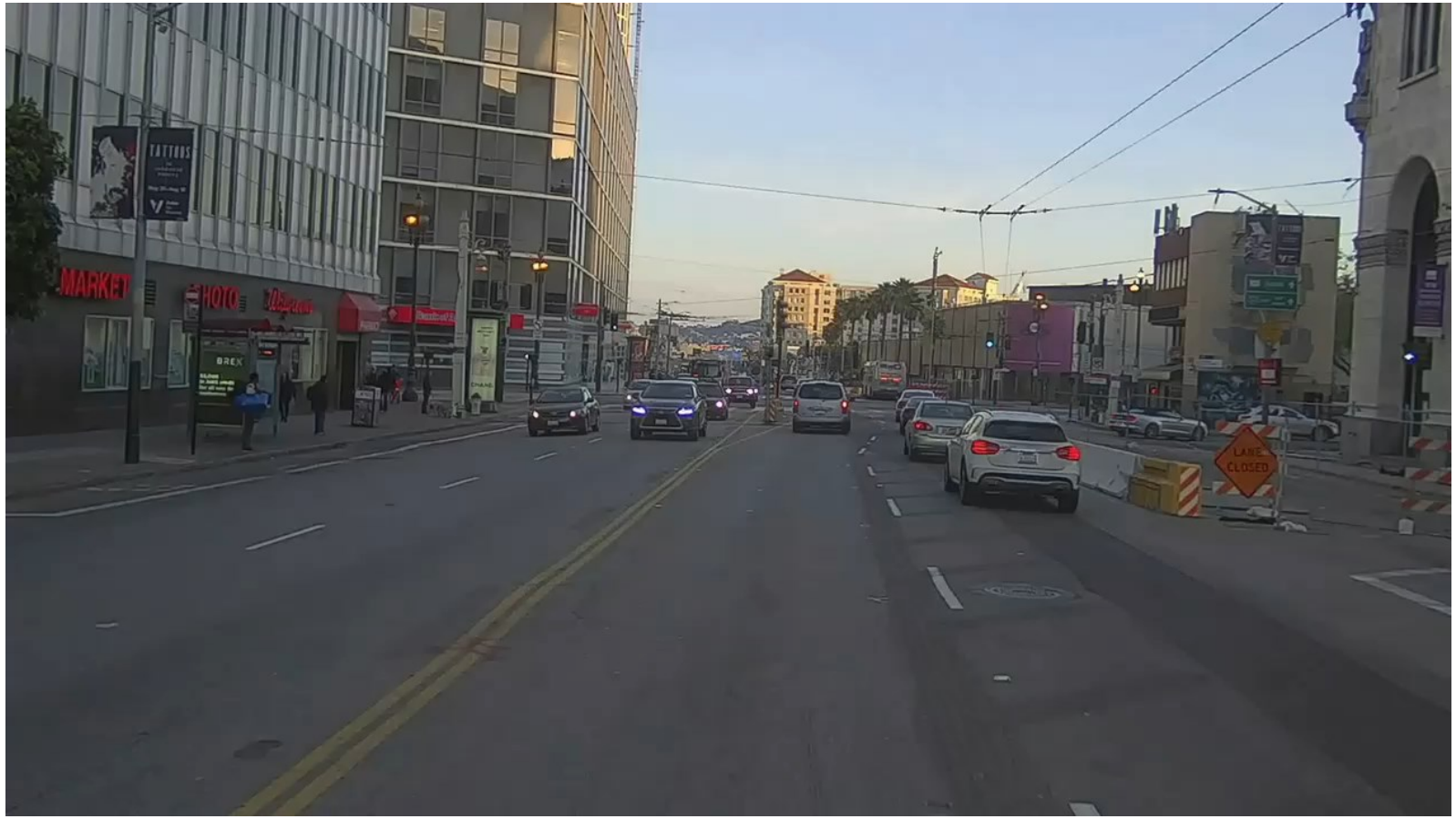
Water-splotches

Eyeglasses

Background problems

Other asymmetries, hair, teeth, Fluorescent bleed

Anything in Any Scene Photorealistic Video Object Insertion



Detecting AI-generated Content

- Passive Detection
  - Key idea: leverage artifacts in AI-generated content
  - High false positives/negatives
  - Abandoned by OpenAI

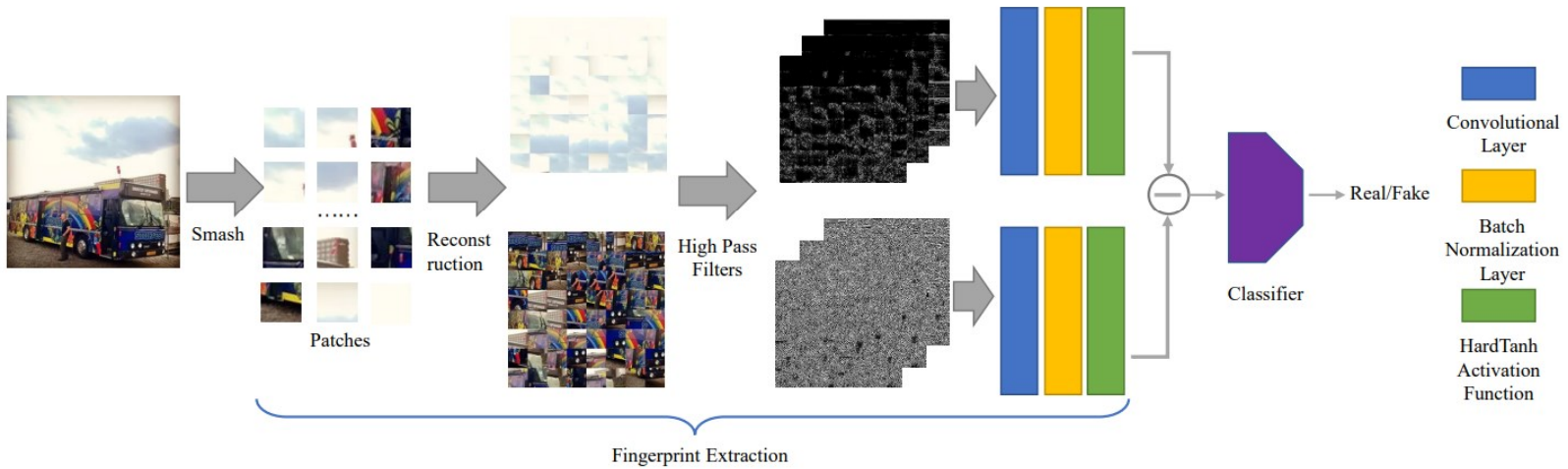
- Watermark-based Detection
  - Deployed by Google, Microsoft, OpenAI, Stability AI, etc.

Passive Detection

- Binary classification: cannot extend to unseen generative models
  - But data augmentation can significantly boost the generalization for the detector.

- Universal fingerprint across different generative models
  - Consistently anomalous behavior of fake images in the frequency domain.
  - Upsampling operation results in the frequency abnormality of fake images.
  - Image generated by a diffusion model are more likely to be accurately reconstructed by a pre-trained diffusion model.
  - So, previous researchers adopt the reconstruction error of an image as the fingerprint to identify diffusion-generated fake images.

PatchCraft: Exploring Texture Patch for Efficient AI-generated Image Detection (1)



- Inter-pixel correlation is determined by its camera device (CMOS) and ISP (Image Signal Processing).
- The inter-pixel correlations of real images between rich and poor texture regions are very close.

PatchCraft: Exploring Texture Patch for Efficient AI-generated Image Detection (2)

- Texture diversity measurement

$$l_{div} = \sum_{i=1}^{M}\sum_{j=1}^{M-1}(|x_{i,j} - x_{i,j+1}|) + \sum_{i=1}^{M-1}\sum_{j=1}^{M}(|x_{i,j} - x_{i+1,j}|$$
$$+ \sum_{i=1}^{M-1}\sum_{j=1}^{M-1}(|x_{i,j} - x_{i+1,j+1}| + \sum_{i=1}^{M-1}\sum_{j=1}^{M-1}(|x_{i+1,j} - x_{i,j+1}|).$$

- High-pass filters are conducive to suppressing semantic information and magnifying the interpixel correlation

PatchCraft: AI-generated Image Detection

Generator	CNNSpot	FreDect	Fusing	GramNet	LNP	LGrad	DIRE-G	DIRE-D	UnivFD	Ours
ProGAN	**100.00**	99.36	**100.00**	99.99	99.67	99.83	95.19	52.75	99.81	**100.00**
StyleGan	90.17	78.02	85.20	87.05	91.75	91.08	83.03	51.31	84.93	**92.77**
BigGAN	71.17	81.97	77.40	67.33	77.75	85.62	70.12	49.70	95.08	**95.80**
CycleGAN	87.62	78.77	87.00	86.07	84.10	86.94	74.19	49.58	**98.33**	70.17
StarGAN	94.60	94.62	97.00	95.05	99.92	99.27	95.47	46.72	95.75	**99.97**
GauGAN	81.42	80.57	77.00	69.35	75.39	78.46	67.79	51.23	**99.47**	71.58
Stylegan2	86.91	66.19	83.30	87.28	**94.64**	85.32	75.31	51.72	74.96	89.55
WFIR	**91.65**	50.75	66.80	86.80	70.85	55.70	58.05	53.30	86.90	85.80
ADM	60.39	63.42	49.00	58.61	84.73	67.15	75.78	**98.25**	66.87	82.17
Glide	58.07	54.13	57.20	54.50	80.52	66.11	71.75	**92.42**	62.46	83.79
Midjourney	51.39	45.87	52.20	50.02	65.55	65.35	58.01	89.45	56.13	**90.12**
SDv1.4	50.57	38.79	51.00	51.70	85.55	63.02	49.74	91.24	63.66	**95.38**
SDv1.5	50.53	39.21	51.40	52.16	85.67	63.67	49.83	91.63	63.49	**95.30**
VQDM	56.46	77.80	55.10	52.86	74.46	72.99	53.68	**91.90**	85.31	88.91
Wukong	51.03	40.30	51.70	50.76	82.06	59.55	54.46	90.90	70.93	**91.07**
DALLE2	50.45	34.70	52.80	49.25	88.75	65.45	66.48	92.45	50.75	**96.60**
Average	70.78	64.03	68.38	68.67	83.84	75.34	68.68	71.53	78.43	**89.31**

PatchCraft (Ours) https://fdmas.github.io/AIGCDetect/data/

This Lecture

- Accountability
- Detecting AI-generated Content
- Watermarking Techniques
- Evading Watermarking-based Detection

Watermarking Techniques

- Creators embed identifiable markers or metadata into AI models, datasets, or outputs to trace their origin or prove ownership.

- Watermarking techniques in AI systems are a potential solution for ensuring accountability, particularly in the context of AI-generated content, intellectual property protection, and responsibility.

AI Model Watermarking

- White-box Watermarking
  - This technique involves embedding the watermark directly into the model's parameters or structure.
  - It requires access to the internal architecture of the model and can be used to verify ownership or identify unauthorized modifications.
- Black-box Watermarking
  - Black-box watermarking embeds a secret, recognizable pattern in the model's outputs rather than its internal structure.
  - It doesn't require access to the internal workings of the AI model and can be verified through observing the model's responses to specific inputs.

Dataset Watermarking

- Perturbation Watermarking
  - This involves making small, controlled modifications (perturbations) to the data, which do not affect its usability but serve as a watermark to prove dataset ownership or trace responsibility.

- Data Label Watermarking
  - A form of watermarking where specific, unique patterns are embedded in the labels of the dataset.
  - This helps identify if a model has been trained on a specific dataset, even when used by unauthorized entities.

Watermarking in Large Language Models

- Token-based Watermarking
  - Inserting specific patterns or sequences of words or tokens into generated text in a way that is hard for a human to detect but easily identified by automated tools.

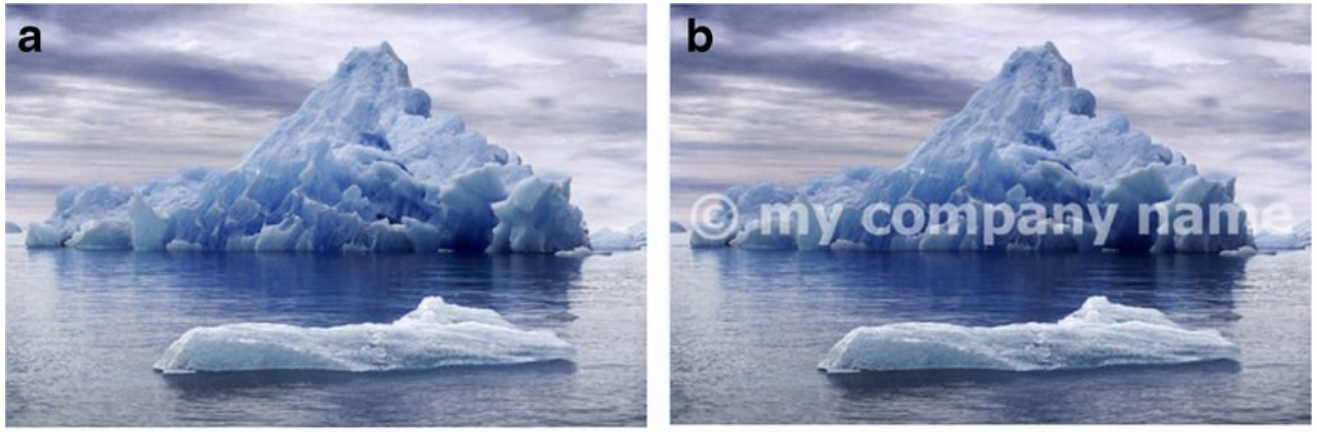
- Sentence Structure Watermarking
  - Subtly altering sentence structures or punctuation in the generated text to leave an identifiable watermark while maintaining the overall meaning.

Watermarking Quality

- Quality. The watermark should be imperceptible to humans, meaning the watermarked version appears identical to the original in raw data modality.
- Generalization. The watermark can be applied to any samples, and the watermarking method is compatible with different models.
- Robustness. The watermark cannot be compromised by existing attack methods.

Digital Watermarking in AI-generated Image

- Invisible Watermarking: embedding markers in a way that is imperceptible to humans but detectable using specific algorithms.
- Visible Watermarking: Placing a visible marker, like a logo or text, onto AI-generated images, videos, or other multimedia content.



Invisible Watermark Visible Watermark

https://link.springer.com/article/10.1007/s11042-013-1515-8

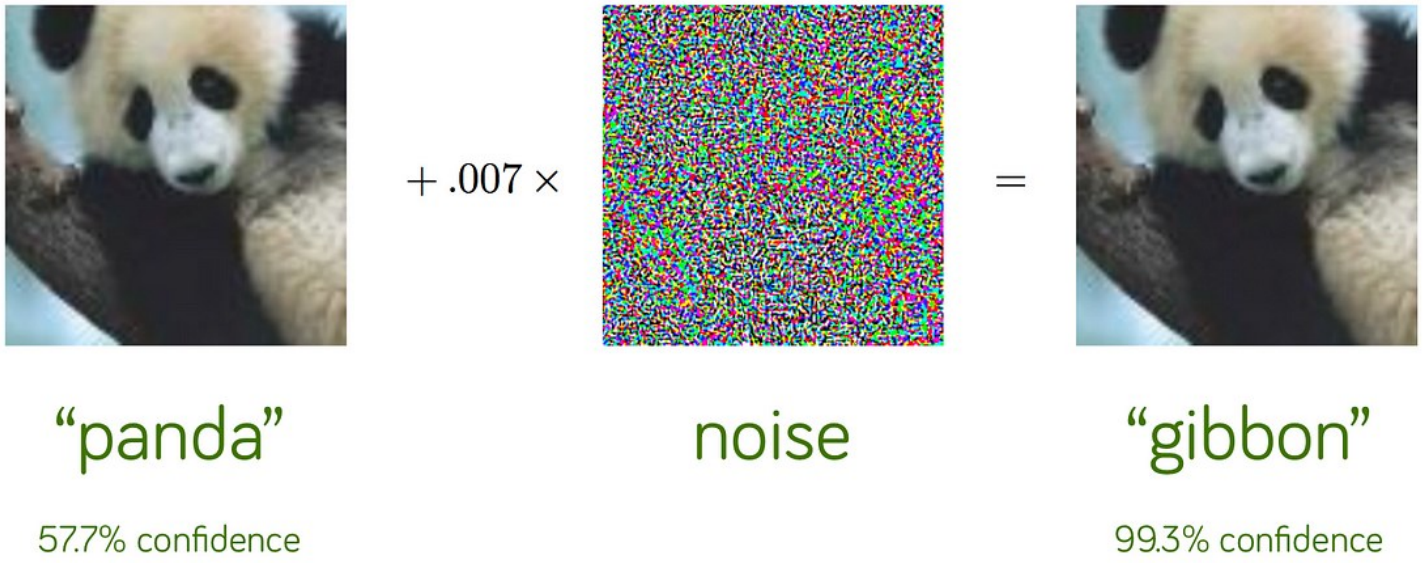
None-learning-based Watermarking Methods

- **dwtDct**: DWT + DCT transform, embed watermark bit into max non-trivial coefficient of block DCT coefficients

- **dwtDctSvd**: DWT + DCT transform, SVD decomposition of each block, embed watermark bit into singular value decomposition

- Discrete wavelet transform (DWT)
- Discrete cosine transform (DCT)
- https://github.com/ShieldMnt/invisible-watermark

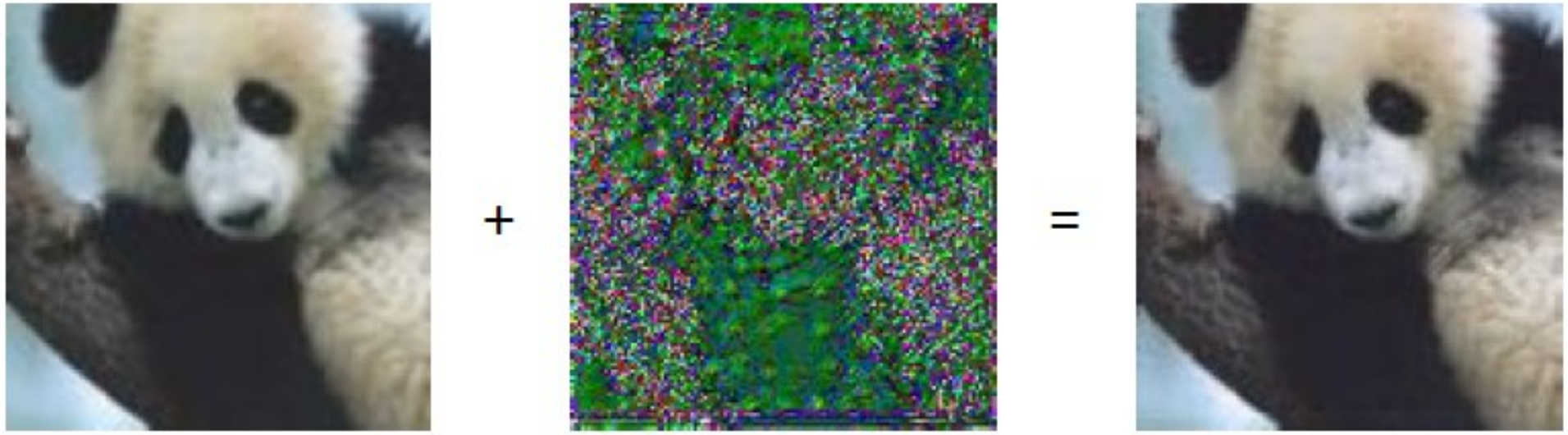
Watermarking and Adversarial Perturbation



Can DNNs extract meaningful information from imperceptible perturbations?

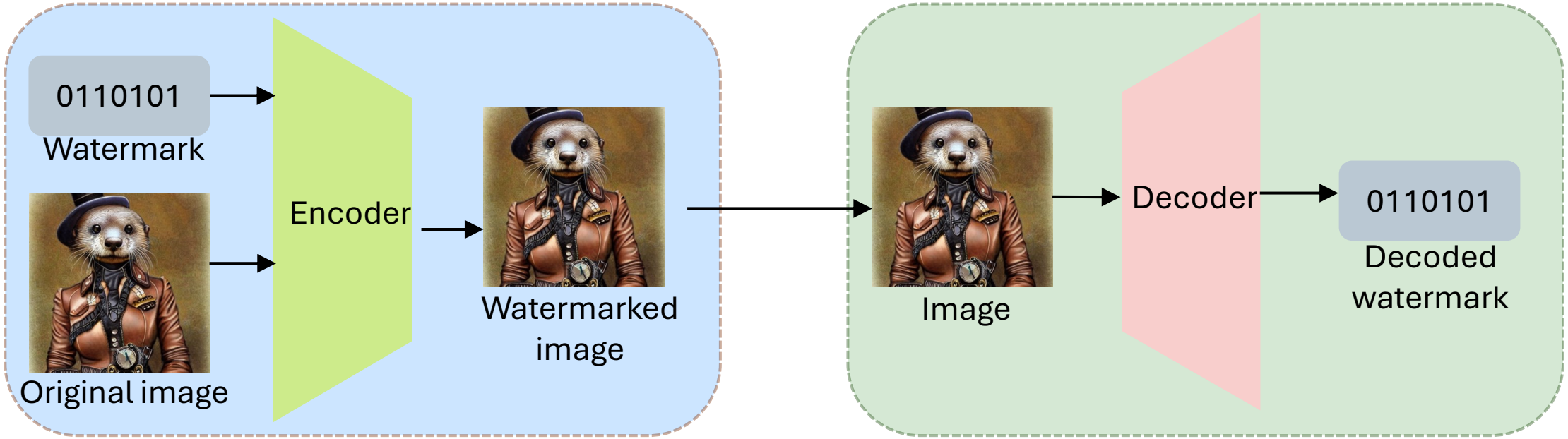
Watermarking and Adversarial Perturbation

Digital watermarking can be used to identify image ownership



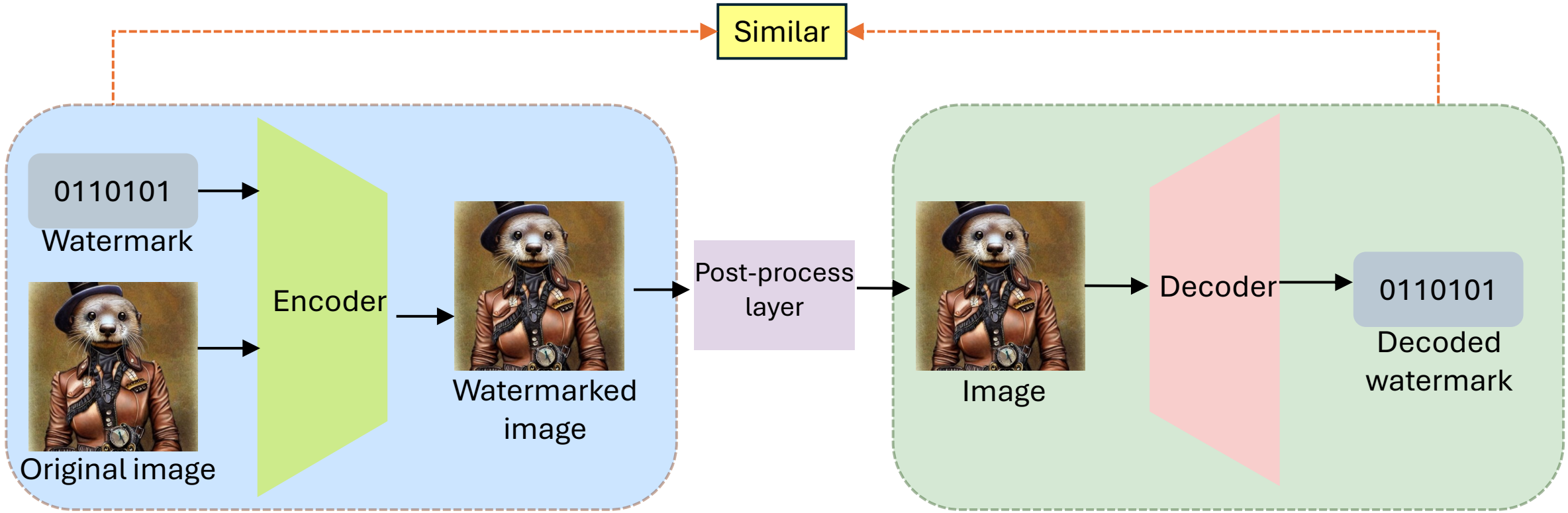
Learning-based Watermarking Methods

Three main components: Watermark (bitstring), Encoder, Decoder



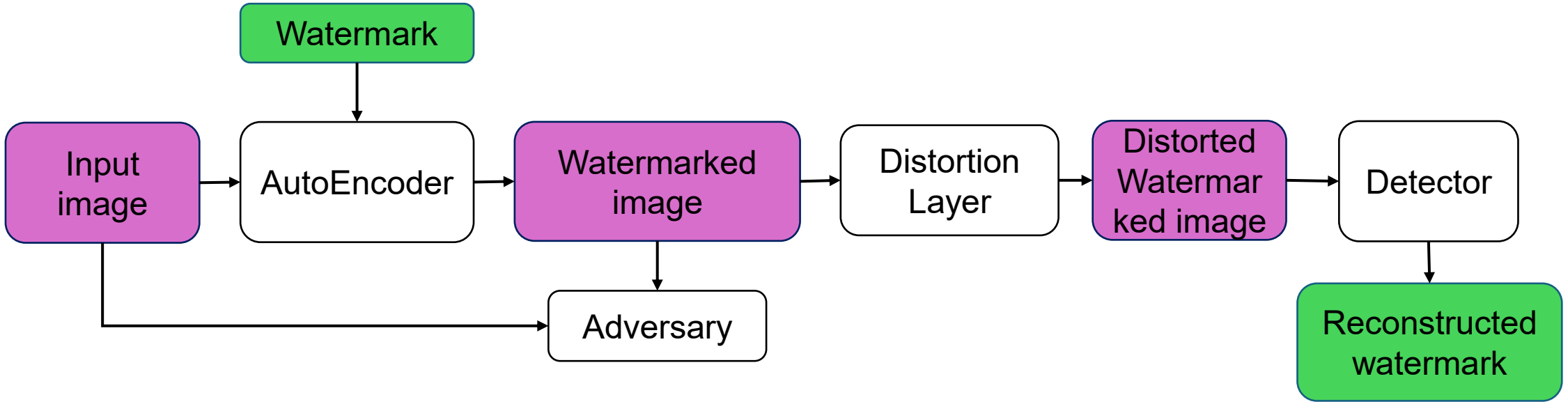
Standard training

Adversarial Training for Watermarking



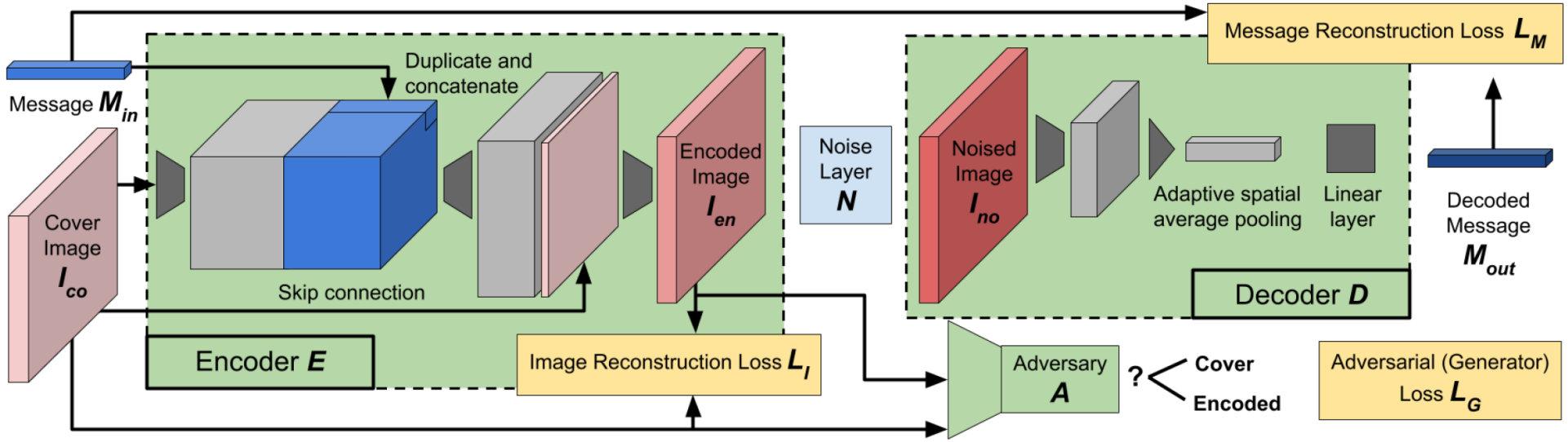
Adversarial training

A General Framework for Training Watermarking Model



- **AutoEncoder.** Embed the watermark into the image, producing an encoded version of it.
- **Adversary.** Improve the quality of the watermarked image and minimize the domain gap between the original and the watermarked image.
- **Detector.** Receive the encoded images or audio and output the extracted message.
- **Distortion Layer.** Simulate potential attack scenarios.

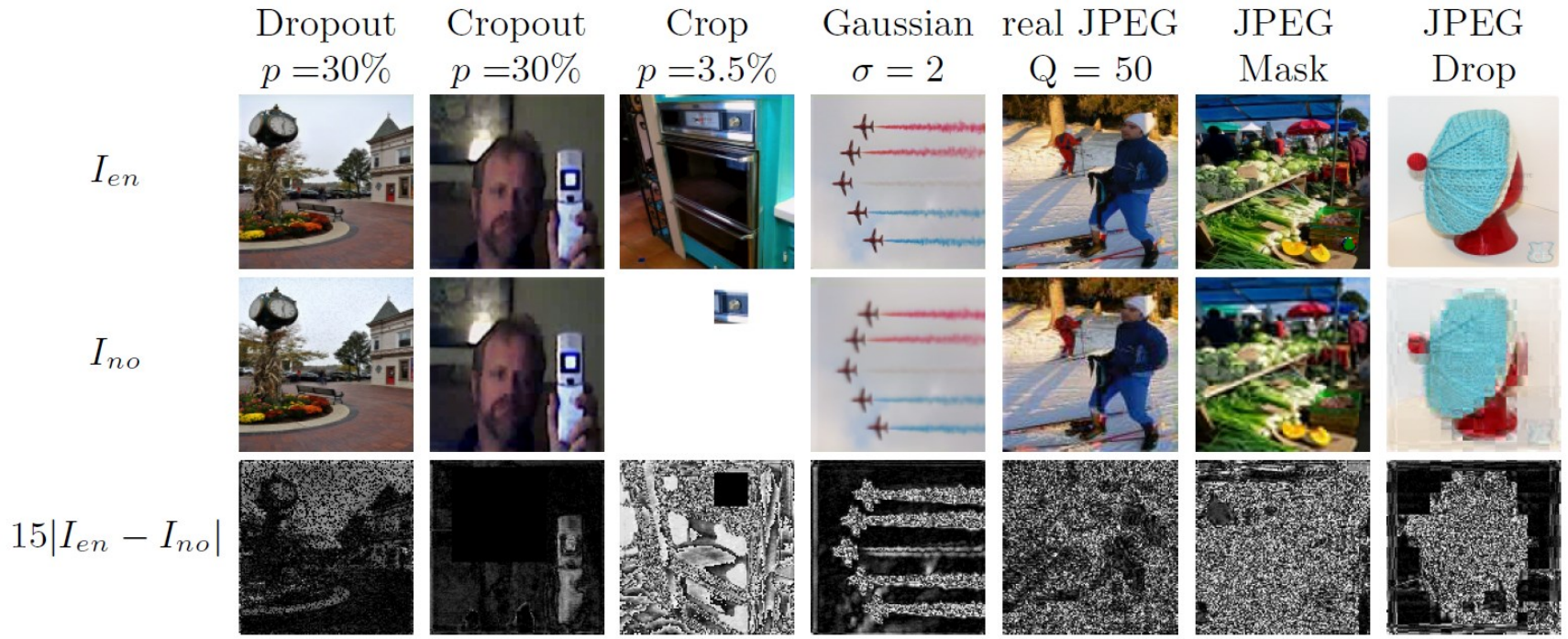
HiDDeN: Hiding Data With Deep Networks



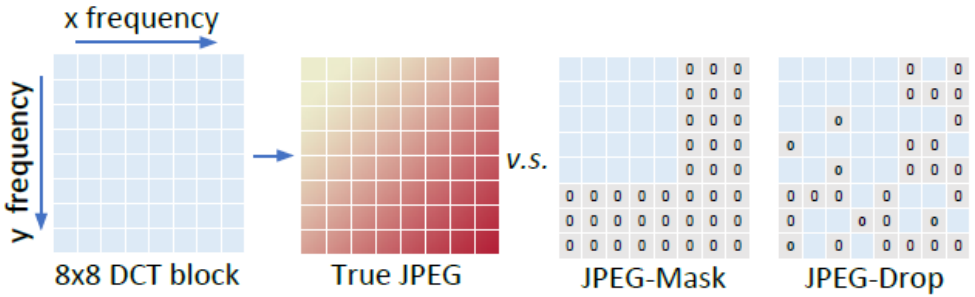
- An **autoencoder** embeds a message into a cover image and outputs an encoded image.
- A **decoder** receives the encoded image and extracts the embedded message.
- The **adversary** predicts whether a given image contains an encoded message.
- The **noisy layer** applies various image transformations to improve robustness against various image distortion scenarios.

https://arxiv.org/pdf/1807.09937

HiDDeN: Distortion Method



- JPEG-Mask zeros a fixed set of high-frequency coefficients
- JPEG-Drop zeros channels with higher drop probabilities for high-frequency coefficients

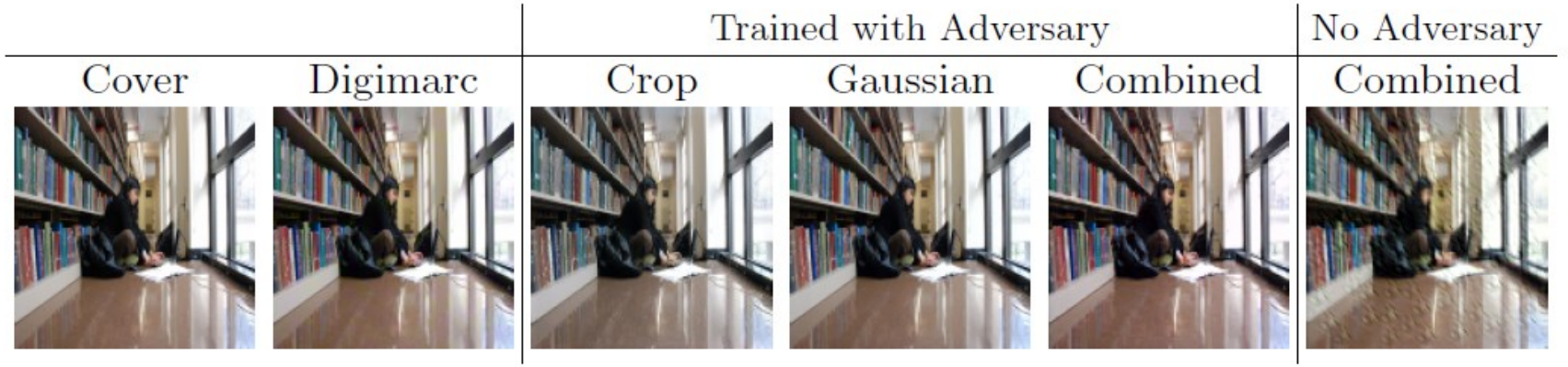


Discrete Cosine Transform (DCT)

HiDDeN: Data Quality Results

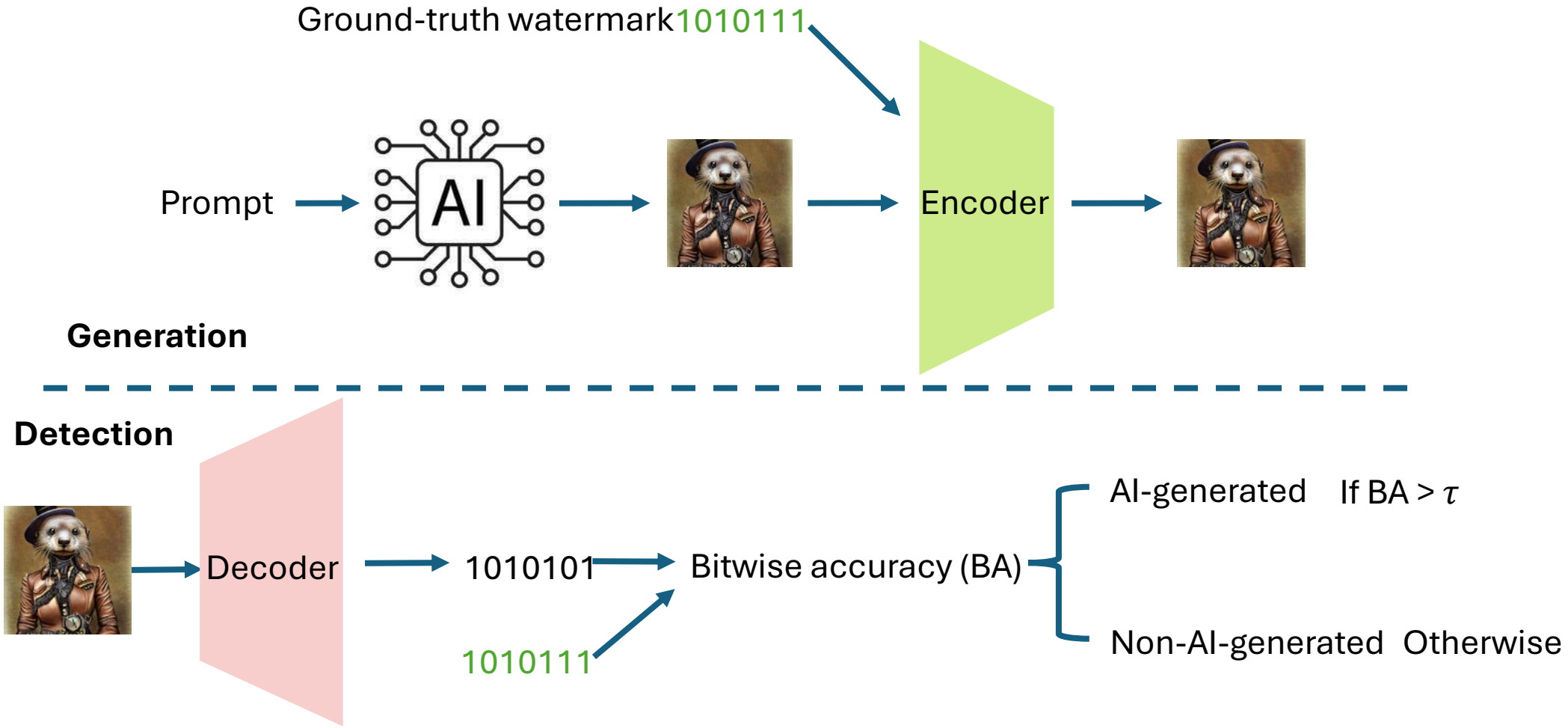
The image distortion between the cover and stego image using peak signal-to-noise ratio (PSNR)

	Digimarc	Identity	Dropout	Cropout	Crop	Gaussian	JPEG-mask	JPEG-drop	Combined
PSNR(Y)	62.12	44.63	42.52	47.24	35.20	40.55	30.09	28.79	33.55
PSNR(U)	38.23	45.44	38.52	40.97	33.31	41.96	35.33	32.51	38.92
PSNR(V)	52.06	46.90	41.05	41.88	35.86	42.88	36.27	33.42	39.38



- Encoded images from the HiDDeN model are **visually indistinguishable** from the cover image
- The model trained against an **adversarial discriminator** produces images with **no visible artifacts**

Watermark-based Detection



How to Set Detection Threshold τ?

Goal: achieve a desired False Positive Rate (FPR)

Non-AI-generated

Decoder

1010101

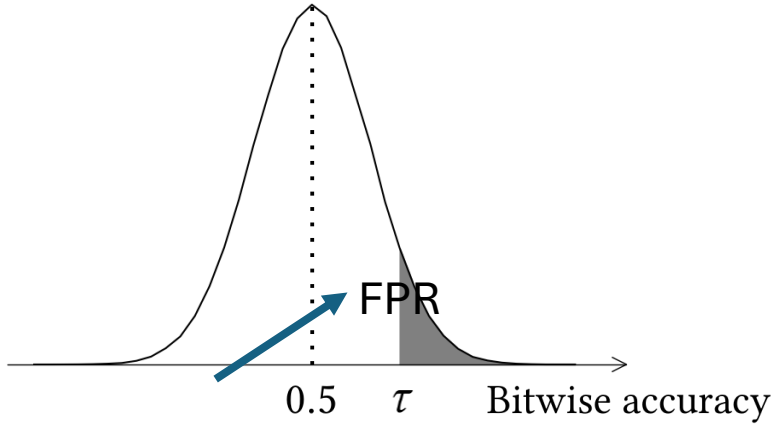
BA

1010111

Uniformly at random

n: watermark length

$$n \cdot BA \sim Binomial\ (n, 0.5)$$



FPR < 10^{-4}

30-bit watermark

$\tau > 0.83$

$$FPR_s(\tau) = \Pr(BA(D(I_o), w) > \tau)$$

$$= \Pr(m > n\tau) = \sum_{k=\lceil n\tau \rceil}^{n} \binom{n}{k} \frac{1}{2^n},$$

https://arxiv.org/abs/2305.03807
https://github.com/zhengyuan-jiang/WEvade

Problem of Single-tail

- Single-tail detector can be easily evaded by simply extending standard adversarial examples to watermarking.

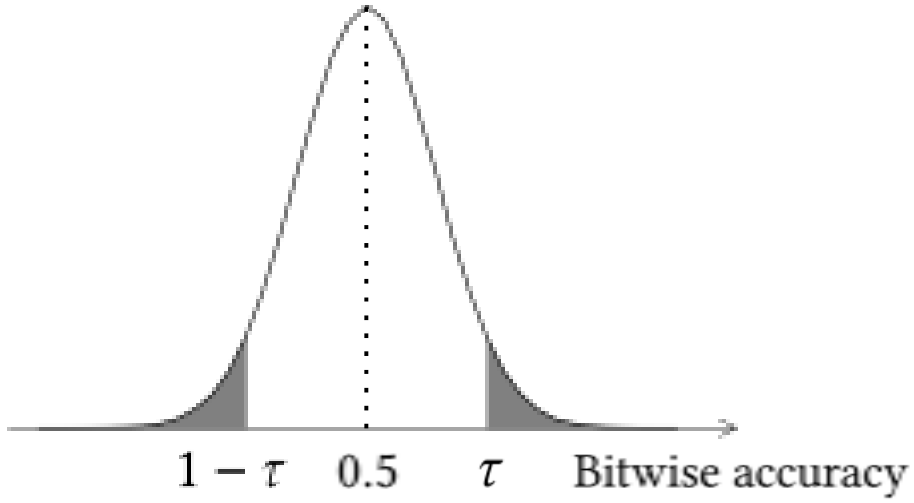
- Adds perturbation to a watermarked image such that the decoded watermark has a very small bitwise accuracy ➔ 0. As a results, this will evade the watermarking-based detection.

Important Observation

- The watermarks decoded from original images have bitwise accuracy close to 0.5, while those decoded from watermarked images have large bitwise accuracy, e.g., close to 1.

- If the bitwise accuracy of the watermark decoded from an image is significantly smaller than 0.5, it is likely to be an adversarial perturbed image.

To avoid evading detection, using double-tail detector.



AI-generated: If BA > τ or BA < 1 - τ

Non-AI-generated: Otherwise

This Lecture

- Accountability
- Detecting AI-generated Content
- Watermarking Techniques
- Evading Watermarking-based Detection

Question: double-tail detector safe to use?

- Intuition: **non-watermarked images have bitwise accuracy ≈ 0.5**
- Attack the detector: add minimal perturbation to make bitwise accuracy ≈ 0.5, making the perturbed watermarked image indistinguishable with non-watermarked ones

$$\min_{\delta} ||\delta||_{\infty}$$

$$s.t.\ |BA(D(I_w + \delta), w) - 0.5| \leq \epsilon$$

Ground-truth watermark: Unknown to attacker

Decoder

Constraints are non-linear, hard to solve!

Solve the Optimization Problem

$$\min_{\delta} ||\delta||_\infty$$
$$s.t.\ |BA(D(I_w + \delta), w) - 0.5| \leq \epsilon$$

$$\min_{\delta} l(D(I_w + \delta), w_t)$$
$$s.t.\ ||\delta||_\infty \leq r,$$
$$BA(D(I_w + \delta), w_t) \geq 1 - \epsilon,$$

Target watermark is a random watermark

Iteratively find the perturbation δ that satisfies the constraints (if possible) for a given r.

Binary search over r to find the smallest perturbation δ that satisfies the constraints

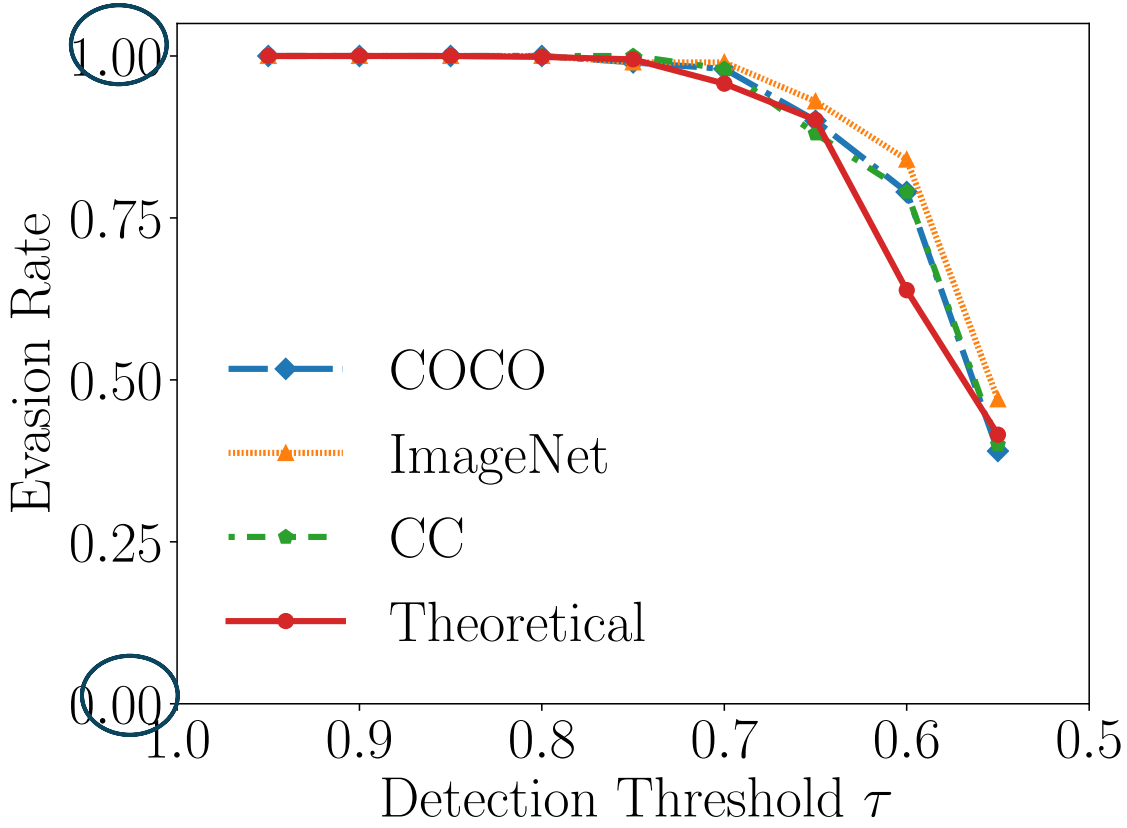
$$[r_a, r_b] \text{ is initialized such that } r_a = 0 \text{ and } r_b \text{ is a large value}$$

W_t bitwise accuracy close to 0.5 compared to any ground-truth watermark

Empirical Evaluation Results

- Evasion rate: probability that a perturbed watermarked image is detected as non-AI-generated

- Detection Threshold:
  - Fraction of bit match accuracy > τ

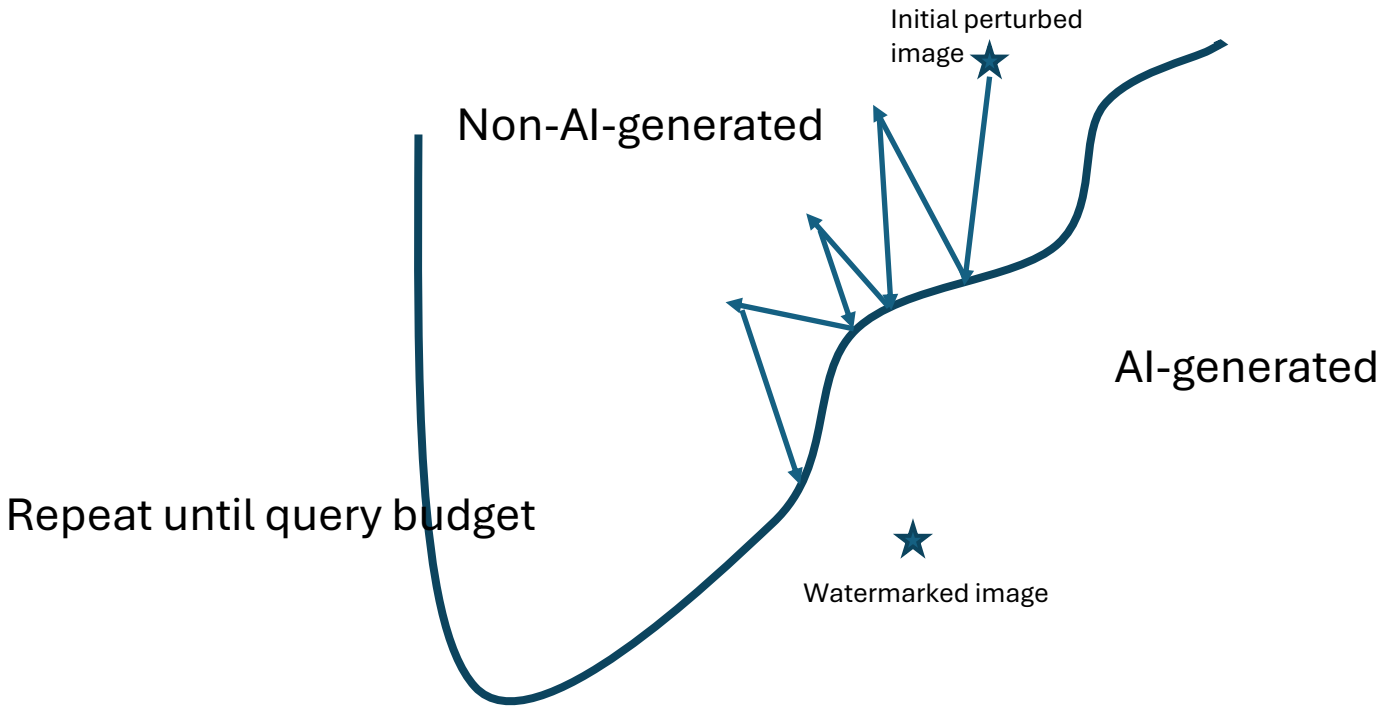


30-bit watermark, standard training

Double-tail detector

Black-box Attack to Evade Watermarking-based Detection

- Surrogate Model
- Query-based



WEvade-B-Q: WEvade-B-Q starts from an initial I_{pw} that evades the target detector. During the iterative process to reduce the perturbation, WEvade-B-Q always guarantees that I_{pw} evades detection.

References

- HiDDeN: Hiding Data With Deep Networks (https://arxiv.org/pdf/1807.09937)

- PatchCraft: Exploring Texture Patch for Efficient AI-generated Image Detection (https://arxiv.org/pdf/2311.12397)

- Evading Watermark based Detection of AI-Generated Content (https://arxiv.org/pdf/2305.03807)