2025.03.24

StegaStamp & 공격 기법에 대한 고찰

국민대학교 이재형
jaehyeong8121@gmail.com

Contents

What is StegaStamp?

사진 속에 보이지 않는 메시지를 삽입하는
딥러닝 기반 **steganography** 시스템

스테가노그래피 : 메시지를 다른 무해한 데이터나 미디어(예: 이미지, 오디오)에 숨겨서 눈에 띄지 않게 전달하는 기술

기존 스테가노그래피

- 디지털 환경만 고려
- 디지털 변형에 견디도록 학습됨
- e.g. LSB 조작, 크롭, JPEG 압축

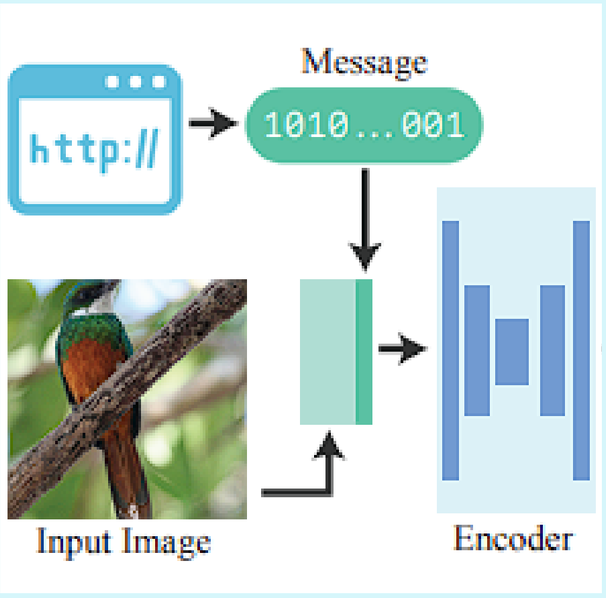
VS

StegaStamp

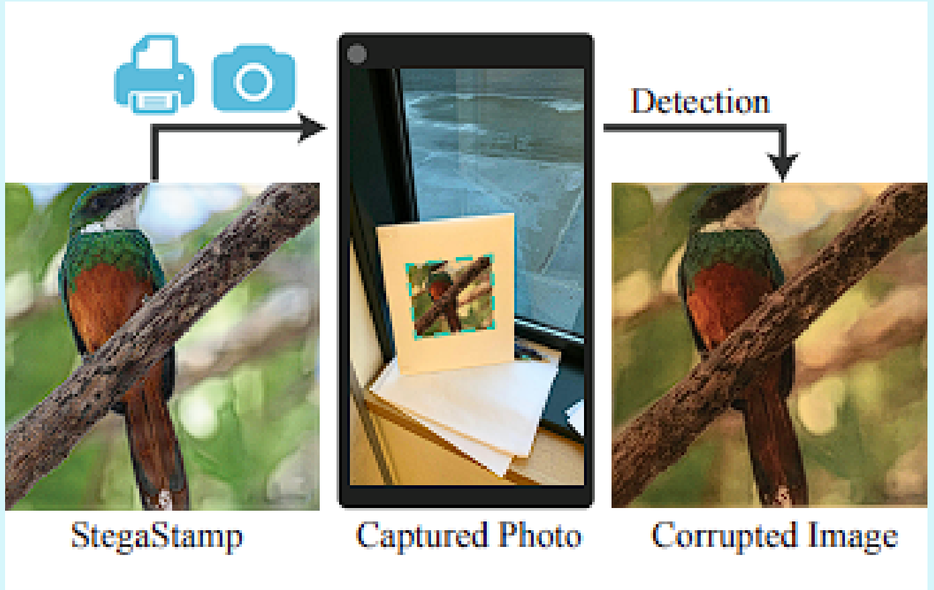
- 실제 이미지 인쇄 및 촬영을 통한 물리적 왜곡 고려
- 물리적 이미지 왜곡을 differentialbe하게 적용
- e.g. 원근 왜곡, 블러

1.1 StegaStamp How it works?

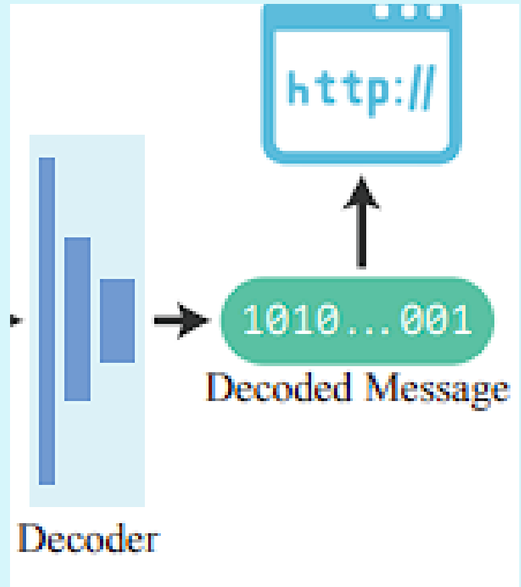
하이퍼링크에 비트 문자열 할당
StegaStamp를 통해
이미지에 문자열 임베드



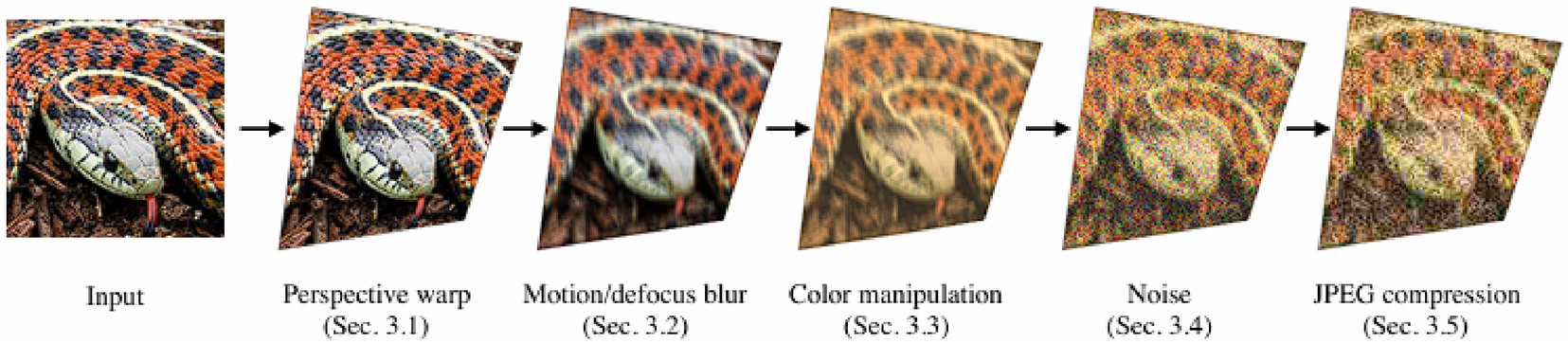
인코딩된 이미지를 물리적으로 인쇄하고
사진을 촬영하여 detector로 식별



StegaStamp decoder로
고유 비트문자열 복원



1.2 StegaStamp Training for Real World Roubstness



실제 환경에서의 왜곡 반영

사진을 촬영할 때 발생할 수 있는
물리적 변형을 모델이 학습할 수 있도록 함

- Perspective warp
- Motion & Defocus Blur
- Color Manipulation
- Noise, JPEG Compression

이미지 복원 능력 향상

이미지 변형을 고려한 학습은 디코더가 변형된 이미지에서 숨겨진 메시지를 정확히 복원할 수 있도록 도와줌

End-to-End 학습

변형을 differentiable하게 처리하면 전체 시스템을 end-to-end로 학습가능
즉, 이미지 인코딩, 변형, 디코딩의 모든 과정을 하나의 네트워크로 학습하게 됨

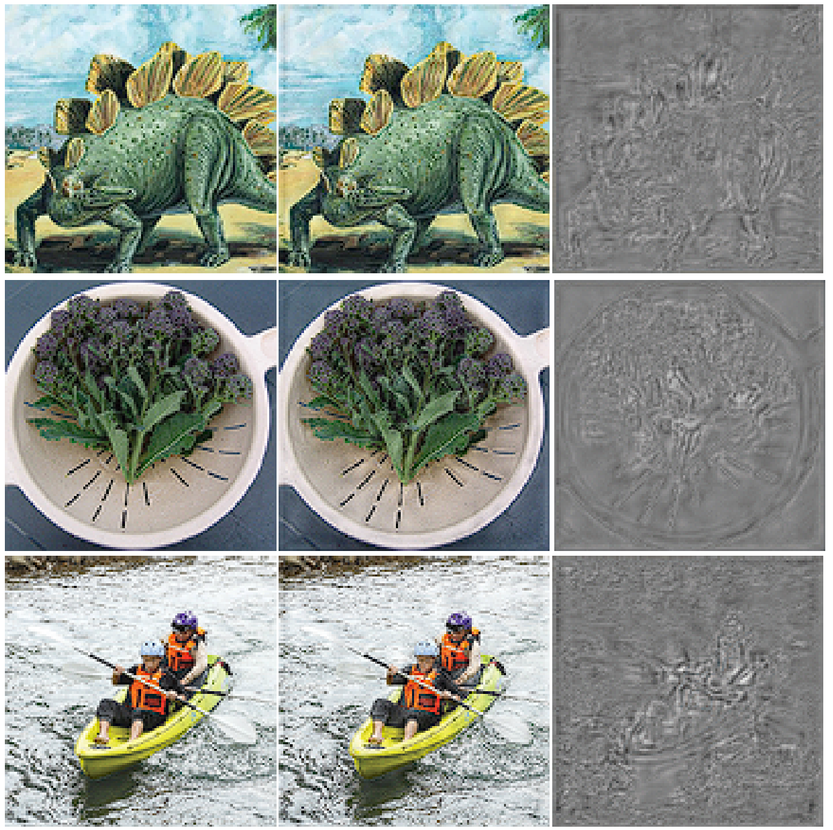
1.3 StegaStamp Implementation Details

1. Encoder

입력 이미지에 100bit 문자열을 임베딩하는 역할

- Input : 400 x 400 RGB 입력 이미지, 삽입할 문자열(1채널, 100bit)
- output : RGB Residual 이미지
- 문자열 삽입 과정 : 100bit -> FCN을 통해 50 x 50 x 3 tensor로 변경
  -> 400 x 400 x 3으로 upsampling

StegaStamp 삽입 이미지 = 원본 이미지 + Residual 이미지



1.3 StegaStamp Implementation Details

2.Decoder

StegaStamp가 삽입된 이미지에서 숨겨진 메시지를 복원하는 역할

이미지가 촬영될 때 발생하는 작은 시점 변화에도 강건하도록 설계

입력
: 워터마킹된 이미지

공간 변환 네트워크 사용
: 촬영될 때 변형을 보정

CNN, Dense Layer 사용
: 숨겨진 메시지 복원

Sigmoid
: 원래 삽입되었던 메시지와 동일한 길이로 복원

3.Detector

StegaStamp가 전체 이미지의 일부분일 가능성이 크다

StegaStamp를 먼저 탐지하고, 위치를 보정하는 과정을 거침

탐지 과정
: 이미지 내 워터마크가 있을 법한 영역을 Segmentation

StegaStamp 위치 보정
: 카메라의 시점 변화로 인한 왜곡을 보정

Convex Hull 알고리즘
: 해당 영역을 감싸는 사각형 검출

1.4 StegaStamp Encoder/Decoder Training Procedure

1. Training Data

- MIR FLICKR 데이터셋
- 400 x 400 Resampling
- bit 문자열 메시지는 훈련 중 무작위로 샘플링되어 이미지와 결합

MIRFLICKR Download

To download, please save the links below. This server supports resumes on downloads.

IMPORTANT: Browsers sometimes have problems to download the collection in one large file without corrupting it. A download manager (Firefox addon *Downthemall* is known to work with large downloads) may help

MIRFLICKR-25000 (released in 2008 aka mirflickr08 or mirflickr25k)

1. Image collection (incl. tags, EXIF), approx 2.9 Gb

Collection in 1 large file (md5: A23D0A8564EE84CDA5622A6C2F947785)

mirflickr25k.zip

2. Annotations

mirflickr25k_annotations_v080.zip

Please refer to README.txt for details.

2. Critic

StegaStamp 모델 학습 과정에서 critic 네트워크를 추가.
이는 이미지 내에서 StegaStamp 메시지가 포함되었는지를 예측하는 역할을 수행

Critic은 StegaStamp의 인코더/디코더와 함께 훈련되며,
두 네트워크는 서로 경쟁하는 방식으로 훈련됨

Critic의 출력은 메시지를 포함하고 있는지, 아닌지를 판별하는 확률값으로, 이를 활용한 Critic Loss가
강건한 인코딩을 가능하게 함

1.4 StegaStamp Encoder/Decoder Training Procedure

3. Loss

$$L = \lambda_R L_R + \lambda_P L_P + \lambda_C L_C + \lambda_M L_M$$

L_R

- L2 Residual Regularization Loss
- 원본 이미지와 StegaStamp가 인코딩된 이미지 간의 픽셀 차이

L_P

- LPIPS Perceptual Loss
- CNN을 통한 이미지 간의 특징 맵을 비교
- 사람이 지각하는 주관적인 품질을 유지하기 위한 Loss

L_C

- Critic 네트워크를 활용하여 StegaStamp가 삽입된 이미지와 원본이 구별되지 않도록 학습

L_M

- 디코더가 숨겨진 메시지를 정확히 복원하도록 하는 Loss

1.4 StegaStamp Encoder/Decoder Training Procedure

4. Optimiztion in the Training Procedure

$$L = \lambda_R L_R + \lambda_P L_P + \lambda_C L_C + \lambda_M L_M$$

가중치 조절

학습 초기에는 디코더가 메시지를 정확히 복원하는 것이 가장 중요하다

$$\lambda_R, \lambda_P, \lambda_C$$

초기에 0으로 설정 후에 점진적으로 증가시키면서 학습

이미지 왜곡 강도 조절

모델이 초반에는 학습하기 쉽도록 왜곡 강도를 0으로 시작하여 점진적으로 증가시키며 학습

이미지 가장자리 발생 패턴 제거

이미지 가장 자리에
L2 Loss 가중치 증가
이를 Cosine Dropoff을 적용하여
가장자리 왜곡을 최소화함

1.5 Real-World & Simulation-Based Evaluation

1. In-the-wild Robustness

- 휴대폰 카메라를 이용한 촬영으로도 정확한 복구 가능
- 또한, StegaStamp 일부가 가려져있어도 강건함
- 실험 결과 경계 사각형이 정확하게 위치하면 디코딩 정확도가 상승됨

1.5 Real-World & Simulation-Based Evaluation

2. Controlled Real-World Experiments

			5th	25th	50th	Mean
Webcam	Printer	Enterprise	88%	94%	98%	95.9%
		Consumer	90%	98%	99%	98.1%
		Pro	97%	99%	100%	99.2%
	Screen	Monitor	94%	98%	99%	98.5%
		Laptop	97%	99%	100%	99.1%
		Cellphone	91%	98%	99%	97.7%
Cellphone	Printer	Enterprise	88%	96%	98%	96.8%
		Consumer	95%	99%	100%	99.0%
		Pro	97%	99%	100%	99.3%
	Screen	Monitor	98%	99%	100%	99.4%
		Laptop	98%	99%	100%	99.7%
		Cellphone	96%	99%	100%	99.2%
DSLR	Printer	Enterprise	86%	96%	99%	97.0%
		Consumer	97%	99%	100%	99.3%
		Pro	98%	99%	100%	99.5%
	Screen	Monitor	99%	100%	100%	99.8%
		Laptop	99%	100%	100%	99.8%
		Cellphone	99%	100%	100%	99.8%

[데이터셋]

- ImageNet 이미지 100개 무작위 선택하여 100bit 메시지를 인코딩하여 StegaStamp 삽입

[테스트 환경]

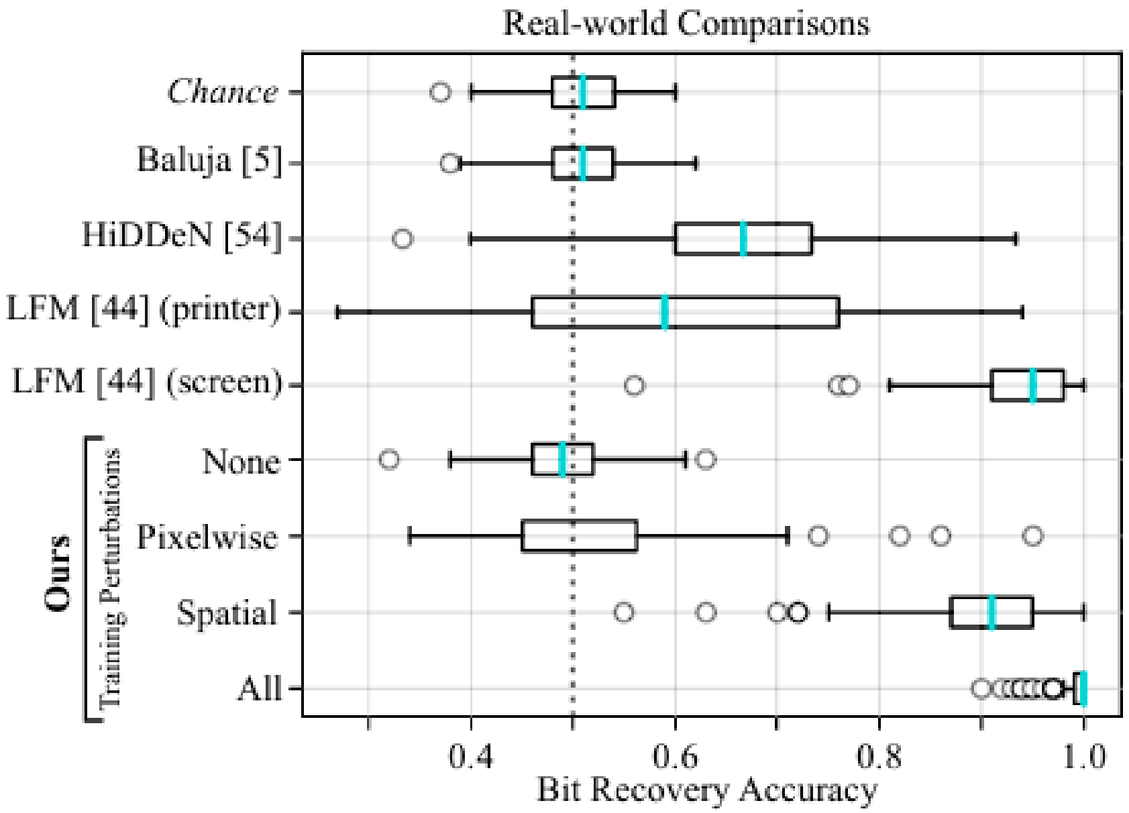
- 3가지 카메라 종류 : Webcam, Cellphone, DSLR
- 각 카메라 종류에 따라 Printer, Screen 2가지로 비교
  - Printer : 물리적 출력 이미지
  - Screen : 디지털 화면에서의 이미지

[정확도 측정 & 결과 해석]

- 정확도는 각 카메라와 디스플레이 방법에 대해 복원된 비트의 정확도를 나타냄
- StegaStamp가 실제 환경에서도 매우 효과적으로 작동함을 증명함

1.5 Real-World & Simulation-Based Evaluation

2. Controlled Real-World Experiments



cellphone camera + consumer printer 조합으로
StegaStamp와 기존 스테가노그래피 모델과 비교하는 실험

[비교 모델]

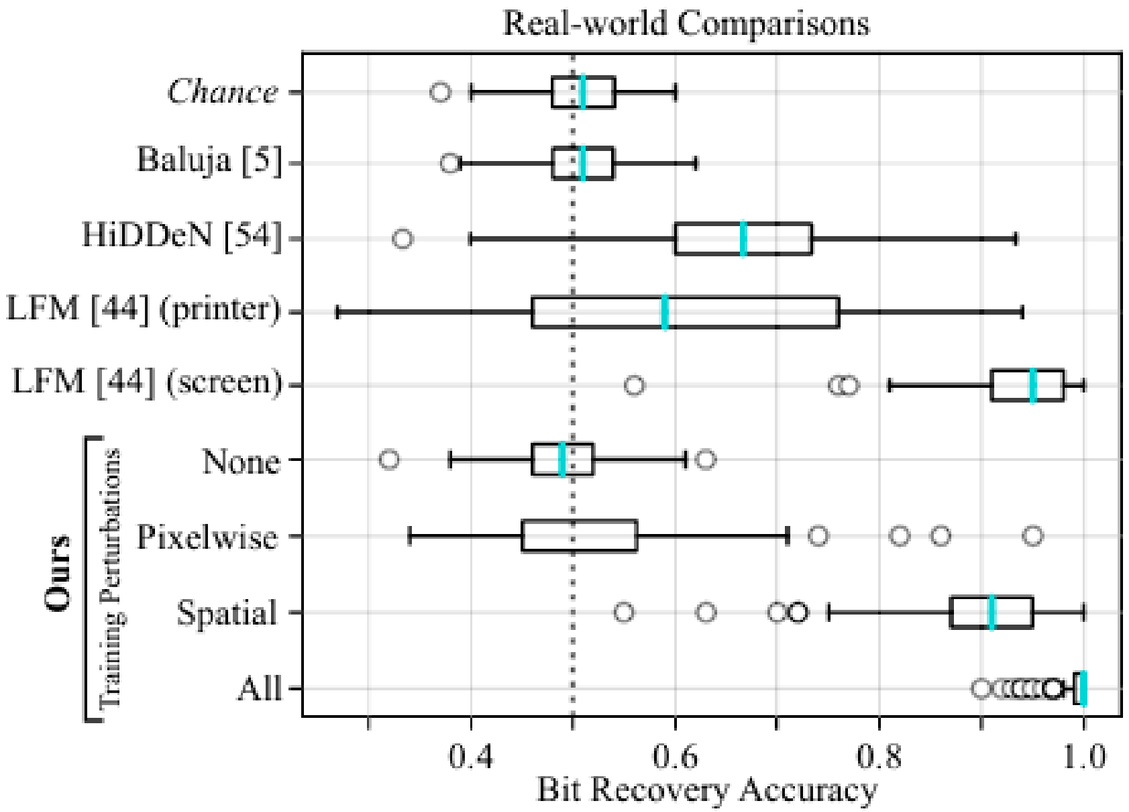
- Baluja : 기본적인 스테가노그래피, 노이즈 증가x
- HiDDeN : 픽셀 단위 변형만 추가하여 훈련됨
- LFM : 디스플레이 환경에서의 노이즈 반영한 모델

[그래프 정보]

- X축 : 이미지에서 숨겨진 메시지를 복구할 수 있는 정도
  - x축 0,5 지점(chance level) : 무작위로 추측할 때 정확도
- Y축 : 훈련 데이터에 적용된 변형 기법의 종류

1.5 Real-World & Simulation-Based Evaluation

2. Controlled Real-World Experiments



[결과 해석]

1.Baluja

- 무작위 추측보다 나은 성능을 보이지 못함
- 훈련 과정에서 노이즈가 추가되지 않았기 때문

2. HiDDeN

- 픽셀 단위만 변형했지만, 실제 환경에서는 여전히 강건하지 못함, 물리적 왜곡(공간 변형)까지 고려하지 않았기 때문

3. LFM

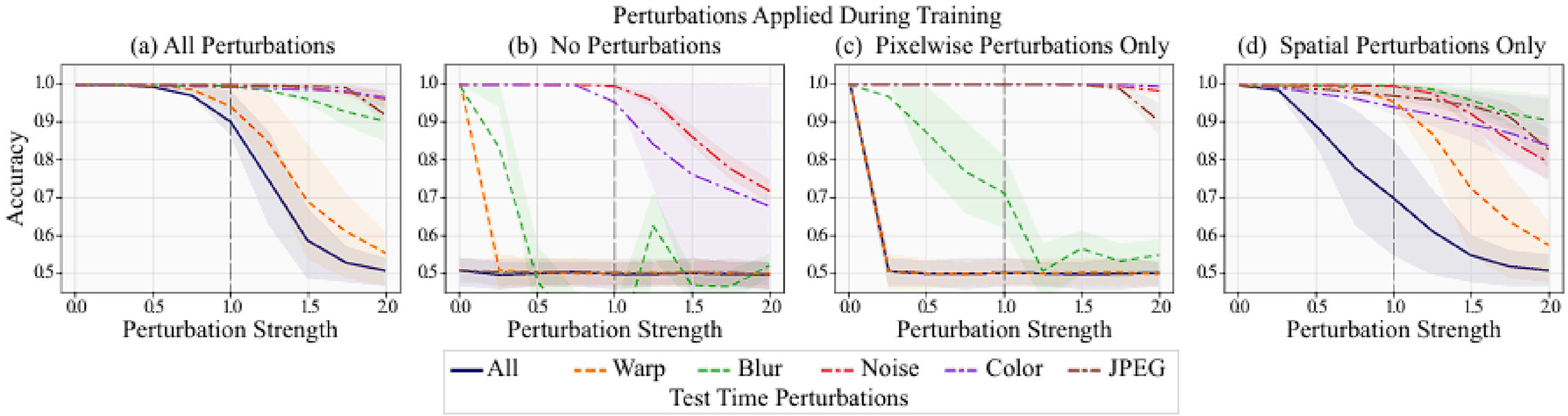
- Screen에서는 성능이 우수하지만, Printer 환경에서는 일반화되지 못함

4. StegaStamp Variants

- Spatial : 공간 변형(왜곡, 회전 등) 추가, 픽셀 단위보다 더 나은 성능을 보임
- **All : 픽셀 단위 + 공간 변형 모두 추가, 가장 강건한 성능**

1.5 Real-World & Simulation-Based Evaluation

3. Synthetic ablation test



[그래프 정보]

- 모델이 훈련 중에 적용된 다양한 이미지 변형 방식이 디코딩 정확도에 미치는 영향을 평가
- X축 : 변형 강도
- Y축 : 디코딩 정확도

[결과 해석]

(a) : 모든 변형 유형에 대해 상대적으로 완만한 정확도 감소를 보임, 가장 강건함

(b) : Warp와 Blur 적용 시 급격한 성능 저하, 현실적인 사용 어려움

(c) : 노이즈와 컬러 변화에 대한 높은 강건성을 보이나, Warp와 Blur에 매우 취약

(d) : Warp와 Blur에 강한 내성을 보이나, 노이즈 및 JPEG 압축에 매우 취약

1.6 Conclusion

1. Limitation

- **고주파 텍스처에서 메시지 삽입의 어려움**
  - StegaStamp는 이미지의 픽셀 값을 조금씩 변형하여 메시지를 삽입
  - 고주파 영역에서는 이러한 미세한 변형이 메시지를 삽입한 흔적으로 눈에 띄게 될 수 있음
  - 정교한 아키텍쳐 및 손실 함수의 개선이 필요
- **구조적 한계**
  - 현재 StegaStamp는 단일 정사각형 이미지에서만 감지 가능하도록 설계됨
  - 더 큰 이미지에 여러 개의 StegaStamp를 자연스럽게 삽입할 수 있다면 활용성이 더 커질 것

2.Significance

- 100비트 하이퍼링크를 자연 이미지에 인코딩하는 종단 간 **딥러닝 프레임워크** 제공
- 이미지 변형 모듈을 통해 **실제 디스플레이-이미지 파이프라인에서 잘 작동**하도록 일반화된 방식을 소개
- 다양한 프린터, 스크린, 카메라 조합에서의 견고한 디코딩 성능을 실험적으로 보여주며, 기존 바코드를 대체할 수 있는 덜 침해적이고 미학적인 방법을 제시

How can i attack StegaStamp?

Invisible Watermarks: Attacks and Robustness

한 가지 워터마킹 기법이 적용된 이미지에 대한 '전체적인' 공격 중심

1. **다중 워터마킹 기법 및 선택적 제거**
2. **국소적 블러 공격**

02 How can i attack StegaStamp?

Baseline model

Tree-Ring, StegaStamp

Attack baseline

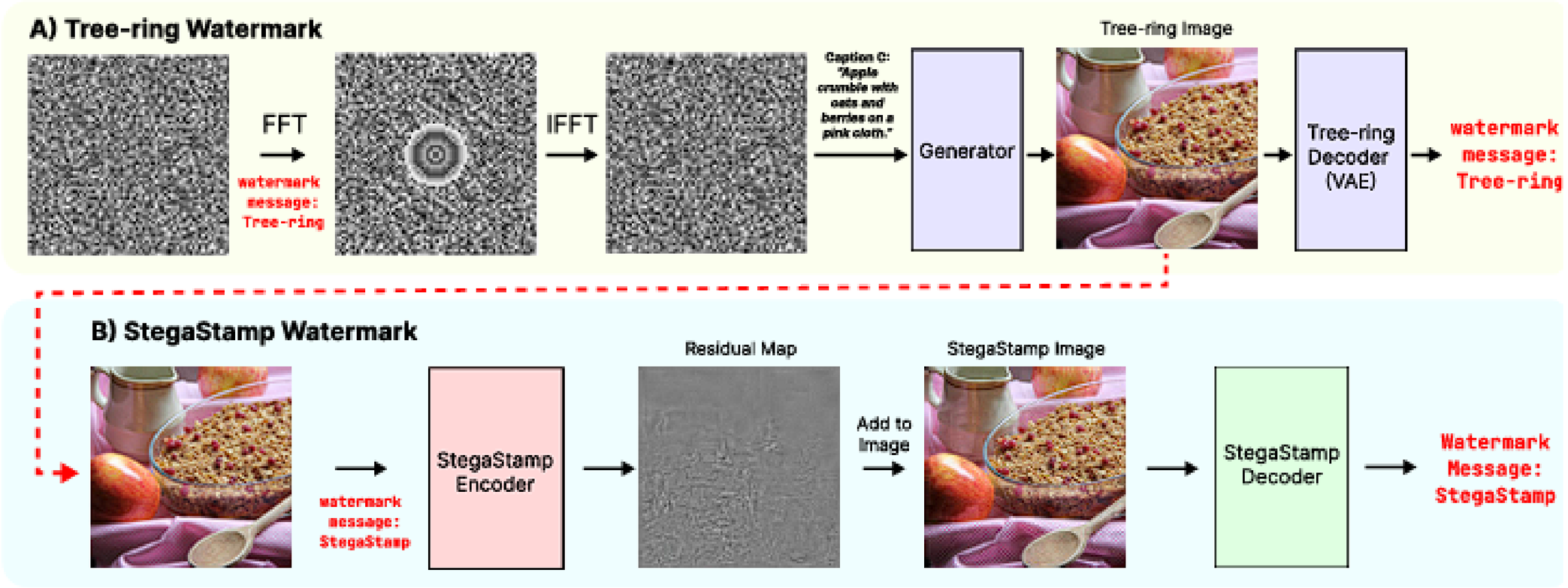
Rotation, Blur(8x8 Gaussian Kernel) -> 워터마크 성능 저하 & 이미지 품질 저하

Regeneration -> 워터마크 성능 저하 & 이미지 품질 유지

1.**여러 워터마킹 기법을 동시에 사용**하여 워터마크 견고성을 개선하는 법을 찾는다

2.대상 이미지의 **대상 영역에 대한 공격을 지역화**하여 이미지 저하를 낮추는 방법을 찾는다

2.1 Naive Stacking Model



- Tree-Ring + StegaStamp
- 이미지 생성 과정에서 Tree-Ring 삽입 -> 생성된 이미지를 StegaStamp Encoder에 넣고 '잔여맵' 생성
  -> 잔여 맵 + 원본 이미지 -> StegaStamp 삽입 이미지 생성

2.1 Naive Stacking Model

Method	Image ℓ_2 distance	Latent ℓ_2 distance
StegaStamp	17.40	118.17
TreeRing	117.58	52.81

[그래프 정보]

- L2 distance : StegaStamp와 Tree-RIng이 각각 적용된 이미지와 원본 이미지 간의 차이
  - Image L2 dist : 픽셀 공간에서 계산된 L2 거리
  - Latent L2 dist : 잠재 공간에서 계산된 L2 거리

[결과 해석]

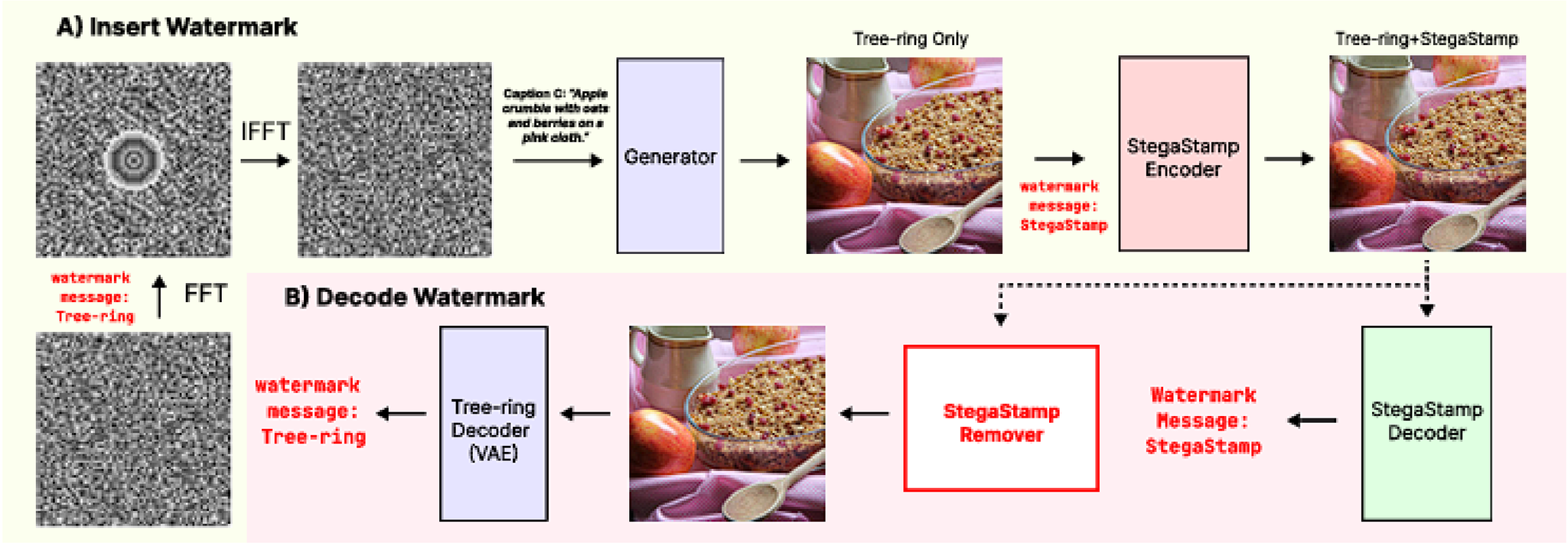
- Tree-Ring은 Image L2 distance에서 더 큰 값을 갖는다 -> 이미지 픽셀 공간에서 더 큰 변화를 일으킴
- StegaStamp는 Latent L2 distance에서 더 큰 값을 갖는다 -> 잠재 공간에서 더 큰 변화를 일으킴
- 즉, StegaStamp와 Tree-Ring은 **서로 다른 특성 공간에서 효과적으로 작동**함을 의미함

2.2 Remover Architecture Stacking Model

- 서로 다른 특성 공간에서 작동하는 Tree-Ring과 StegaStamp를 순차적으로 stack하면
  - Tree-Ring이 먼저 픽셀 공간에서 메시지를 삽입한다
  - 그 후 StegaStamp가 추가될 때 latent space에서 워터마크를 삽입한다
  - 이때, StegaStamp가 latent space서 작동하는 동안 이미지의 고차원적 표현을 변화시키는데,
    이는 **최종적으로 픽셀 공간에 영향을 미치게 되어 Tree-Ring 워터마크의 픽셀 기반 정보가 변형될 수 있다**

- 이러한 Naive Stacking 문제점 극복 방안으로 **Remover Network**를 제안한다
  - Tree-Ring 삽입 -> StegaStamp 추가 삽입 시에 Tree-Ring이 훼손된다
  - 이 때, StegaStamp가 삽입된 이미지에서 **StegaStamp를 제거하여 Tree-Ring 워터마크를 유지하도록** 하는 역할을 하는 Remover Network 도입한다
  - 이는 처음 Tree-Ring이 적용된 이미지와 가깝게 복원되는 것을 목표로 한다

2.2 Remover Architecture Stacking Model



[목적]

- StegaStamp를 제거하고 Tree-Ring 워터마크가 손상되지 않도록 원래 Tree-Ring이 적용된 이미지와 가깝게 복원

2.2 Remover Architecture Stacking Model

[알고리즘]

1. Tree-Ring 워터마크 삽입

$$I_{FFT} \leftarrow FTT(I)$$
$$I_{TR-FFT} \leftarrow Embed(I_{FFT}, M_{TR})$$
$$I_{TR} \leftarrow IFFT(I_{TR-FFT})$$

2. StegaStamp 생성

$$I_{TR+SS} \leftarrow Encoder_{SS}(I_{TR}, M_{SS})$$

3. StegaStamp 디코딩

$$M_{SS} \leftarrow Decoder_{SS}(I_{TR+SS})$$

4. StegaStamp 제거

$$I_{TR-Removed} \leftarrow Remover_{SS}(I_{TR+SS})$$

5. Tree-Ring 디코딩

$$I_{TR-Removed-FFT} \leftarrow FFT(I_{TR-Removed})$$
$$M_{TR} \leftarrow Decoder_{TR}(I_{TR-Removed-FFT})$$

[손실함수]

$$\mathcal{L}_{\text{remover}} = \frac{1}{N} \sum_{i=1}^{N} \| I_{\text{TR\_Removed}}^{(i)} - I_{\text{TR}}^{(i)} \|_2^2,$$

- StegaStamp 제거 네트워크의 출력을 Tree-Ring이 적용된 원본 이미지와 가깝도록 하는 L2 거리 기반 손실함수

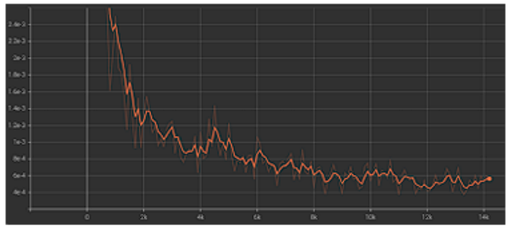


Figure 3: **Loss plot when training the stegastamp remover network.**

Table 6: **Final training loss and validation loss.**

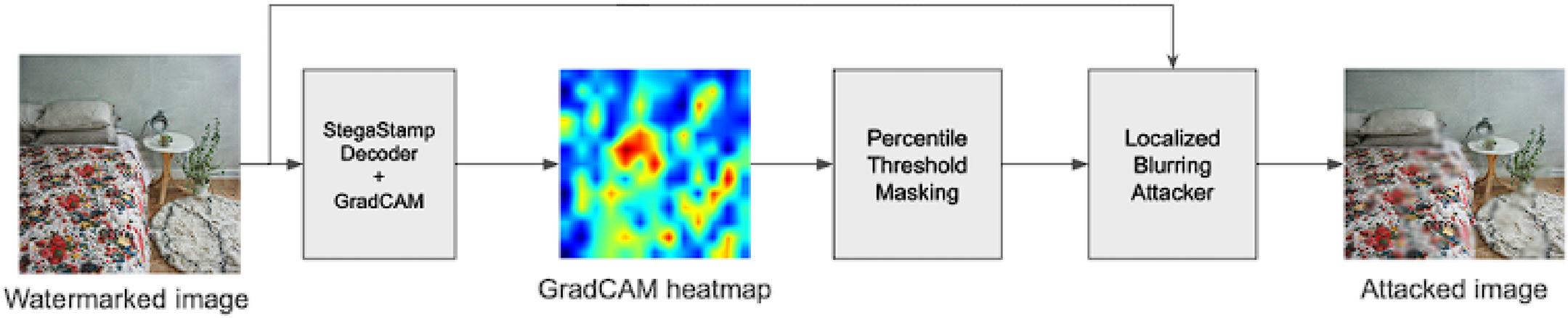
	Training Loss	Validation Loss
Remover Network	0.00057	0.00108

제안된 Remover Network는 StegaStamp를 제거하면서도 Tree-Ring 워터마크를 유지할 수 있도록 학습된다

2.3 Localized Blurring Attack (LBA)

- 위 사진은 원본 이미지, StegaStamp 적용 이미지, Residual 이미지
- Residual 이미지를 보면
  - 워터마크가 적용된 부분은 '고대비(high contrast)'를 갖는 요소들(카약, 사람들)이다
  - 평탄(flat)한 부분보다는 시각적으로 두드러지는 곳에 워터마킹이 집중됨
- LBA 공격은 이러한 **특정 중요 영역만 공격**하며, 품질 저하를 최소화하는 것을 목표로 함

2.3 Localized Blurring Attack (LBA)



StegaStamp Decoder + GradCAM

StegaStamp Decoder와 GradCAM을 이용하여 워터마크 복원 시 네트워크가 **가장 집중하는 영역을 색상 히트맵으로 변환**

Percentile Threshold Masking

특정 임계값 이상을 갖는 영역을 설정하고, Binary Mask를 생성

Localized Blurring Attack 적용

마스크에 **해당하는 영역을 blur처리하여 워터마크를 효과적으로 제거**하여 전체 Blur보다 시각적 품질 저하 방지

Results & Limitations

3.1 Combining StegaStamp and Tree-Ring

Method	Bit Accuracy
StegaStamp	0.997
Naively Stacking	0.998

Metric	Tree-Ring	+ Stacked	+ Remover
AUC	0.9956	0.9936	0.9950
TPR@1%FPR	0.96	0.95	0.96

Naively Stacking(Remover Network 포함)은
개별 속성을 방해하지 않고 상호 보완적이다
이는 stacking의 견고성, 다층 워터마킹 시스템
설계의 잠재력을 강조한다

일반적인 Stacked보다 Remover
Network를 사용함으로써 성능 저하 없이
Tree-Ring을 원본 성능에 가까운 AUC와
TPR@1%FPR을 얻을 수 있다

3.1 Combining StegaStamp and Tree-Ring

Table 9: Detection rate for different methods under various transformations.

Method	None	Blurring	Rotation
Original Tree-ring	0.911	0.218	0.000
Original StegaStamp	1.000	0.089	0.010
Naively Stacking	1.000	0.158	0.000
Remover	1.000	0.168	0.000

- Table 9는 공격 환경에서의 각각의 **Detection Rate(검출율)**을 나타낸다
- None(변형 없음) : StegaStamp > Tree-Ring
- Blurring : Tree-Ring > Remover > Naively Stacking > StegaStamp : StegaStamp가 유난히 취약함
- Rotation : 회전에 대해서는 모든 방법이 탐지하지 못함

->> **Blurring에 대해서는 Remover, Stacking이 StegaStamp를 개별적으로 사용할 때보다 성능 향상은 있지만 여전히 부족함, 회전 변형에 대해서는 어떤 방법도 유효하지 않음**

3.2 Localized Blurring Attack

Table 10: Detection Rates and FIDs for Different Attacks and Configurations

Attack Type	Percentile	Blur Kernel Size (Detection Rates)			Blur Kernel Size (FID)			Regeneration Strength 60	
		5	11	31	5	11	31	Detection Rate	FID
Localized Blurring (LBA)	0	1.0000	0.2594	0.0366	21.9822	50.1141	118.6756	-	-
	25	1.0000	0.4908	0.0376	21.6435	54.2929	136.1907	-	-
	50	1.0000	0.8536	0.0554	18.8150	40.4577	88.0998	-	-
	75	1.0000	0.9980	0.5372	12.1990	20.4747	36.0739	-	-
Randomized Attack	0	1.0000	1.0000	0.9874	7.8892	11.4521	17.2926	-	-
	25	1.0000	1.0000	0.9882	8.5146	13.2648	20.3889	-	-
	50	1.0000	1.0000	0.9796	9.6937	16.6092	25.4361	-	-
	75	1.0000	1.0000	0.8936	11.3115	21.3751	33.2434	-	-
Straight Blurring	-	1.0000	0.1014	0.0000	26.97	62.39	150.24	-	-
Baseline Regeneration	-	-	-	-	-	-	-	0.0100	16.3

- Table 10은 LBA를 포함한 여러 공격 유형에 대한 **Detection Rates**와 **FID**를 비교한 결과이다
  - 또한, 다양한 Blur 커널 크기와 LBA 공격의 임계값에 따라 성능 차이를 분석한다
- **공격 유형**은 LBA, Randomized Attack(무작위로 블러 공격), Straight Blurring, Regeneration이다
- **LBA 공격 강도**는 0, 25, 50, 75으로 나누고, **Blur 커널 크기**는 5, 11, 31로 나누어 분석한다

3.2.1 LBA

Table 10: Detection Rates and FIDs for Different Attacks and Configurations

Attack Type	Percentile	Blur Kernel Size (Detection Rates)			Blur Kernel Size (FID)			Regeneration Strength 60	
		5	11	31	5	11	31	Detection Rate	FID
Localized Blurring (LBA)	0	1.0000	0.2594	0.0366	21.9822	50.1141	118.6756	-	-
	25	1.0000	0.4908	0.0376	21.6435	54.2929	136.1907	-	-
	50	1.0000	0.8536	0.0554	18.8150	40.4577	88.0998	-	-
	75	1.0000	0.9980	0.5372	12.1990	20.4747	36.0739	-	-

- FID : 이미지 품질 평가 척도, 낮을수록 품질이 좋음
- Blur kernel size 31
  - Percentile : 50 -> Detection Rates : 0.0554, FID : 88.0998
  - Percentile : 75 -> Detection Rates : 0.5372, FID : 36.0739
- Percentile이 커질수록 더 중요한 부분만 선택하여 정밀한 Blur 공격을 하므로, Blurring을 적용하는 영역은 더 작아지므로 이미지 품질 저하 정도는 감소한다 ->> FID 감소
- 그러나 검출율이 0.5372로 크게 증가함 ->> **LBA 공격 효과 감소**

3.2.2 Randomized Attack

Table 10: Detection Rates and FIDs for Different Attacks and Configurations

Attack Type	Percentile	Blur Kernel Size (Detection Rates)			Blur Kernel Size (FID)			Regeneration Strength 60	
		5	11	31	5	11	31	Detection Rate	FID
Localized Blurring (LBA)	0	1.0000	0.2594	0.0366	21.9822	50.1141	118.6756	-	-
	25	1.0000	0.4908	0.0376	21.6435	54.2929	136.1907	-	-
	50	1.0000	0.8536	0.0554	18.8150	40.4577	88.0998	-	-
	75	1.0000	0.9980	0.5372	12.1990	20.4747	36.0739	-	-
Randomized Attack	0	1.0000	1.0000	0.9874	7.8892	11.4521	17.2926	-	-
	25	1.0000	1.0000	0.9882	8.5146	13.2648	20.3889	-	-
	50	1.0000	1.0000	0.9796	9.6937	16.6092	25.4361	-	-
	75	1.0000	1.0000	0.8936	11.3115	21.3751	33.2434	-	-

- Randomized Attack은 무작위로 Blur를 적용하는 공격이다
- 전체적으로 탐지율이 크게 줄지 않고, FID도 높지 않다 ->> 의미있는 공격 효과가 보이지 않음
- 중요한 픽셀을 표적으로 삼아 효과적으로 공격하는 LBA와 다르게, **Randomized는 의미 없는 영역을 Blurring하는 것이므로 효과가 미미**함

3.2.3 Straight Blurring

Table 10: Detection Rates and FIDs for Different Attacks and Configurations

Attack Type	Percentile	Blur Kernel Size (Detection Rates)			Blur Kernel Size (FID)			Regeneration Strength 60	
		5	11	31	5	11	31	Detection Rate	FID
Localized Blurring (LBA)	0	1.0000	0.2594	0.0366	21.9822	50.1141	118.6756	-	-
	25	1.0000	0.4908	0.0376	21.6435	54.2929	136.1907	-	-
	50	1.0000	0.8536	0.0554	18.8150	40.4577	88.0998	-	-
	75	1.0000	0.9980	0.5372	12.1990	20.4747	36.0739	-	-
Randomized Attack	0	1.0000	1.0000	0.9874	7.8892	11.4521	17.2926	-	-
	25	1.0000	1.0000	0.9882	8.5146	13.2648	20.3889	-	-
	50	1.0000	1.0000	0.9796	9.6937	16.6092	25.4361	-	-
	75	1.0000	1.0000	0.8936	11.3115	21.3751	33.2434	-	-
Straight Blurring	-	1.0000	0.1014	0.0000	26.97	62.39	150.24	-	-

- Straight Blurring은 이미지 전체 영역에 동일한 Blurring을 적용하는 공격이다
- Blur Kernel Size가 31일 때
  - 검출율 : 0 -> StegaStamp 완벽 제거함
  - FID : 150.24 -> 매우 높아서 이미지 품질이 심각하게 훼손됨
- **현실적인 사용이 어렵다**

3.2.4 Baseline Regeneration

Table 10: Detection Rates and FIDs for Different Attacks and Configurations

Attack Type	Percentile	Blur Kernel Size (Detection Rates)			Blur Kernel Size (FID)			Regeneration Strength 60	
		5	11	31	5	11	31	Detection Rate	FID
Localized Blurring (LBA)	0	1.0000	0.2594	0.0366	21.9822	50.1141	118.6756	-	-
	25	1.0000	0.4908	0.0376	21.6435	54.2929	136.1907	-	-
	50	1.0000	0.8536	0.0554	18.8150	40.4577	88.0998	-	-
	75	1.0000	0.9980	0.5372	12.1990	20.4747	36.0739	-	-
Randomized Attack	0	1.0000	1.0000	0.9874	7.8892	11.4521	17.2926	-	-
	25	1.0000	1.0000	0.9882	8.5146	13.2648	20.3889	-	-
	50	1.0000	1.0000	0.9796	9.6937	16.6092	25.4361	-	-
	75	1.0000	1.0000	0.8936	11.3115	21.3751	33.2434	-	-
Straight Blurring	-	1.0000	0.1014	0.0000	26.97	62.39	150.24	-	-
Baseline Regeneration	-	-	-	-	-	-	-	0.0100	16.3

- Regeneration 공격은 탐지율이 0.0100, FID 16.3으로 StegaStamp를 거의 완벽히 제거함과 동시에, 이미지 품질이 유지된다
- **Regeneration 공격이 StegaStamp를 가장 효과적으로 공격하는 방법**이다

3.3 Limitations

- **LBA**의 한계
  - Blurring 자체가 본질적으로 픽셀 정보를 손상시키는 공격이므로, **재생성 공격에 비해 이미지 품질을 유지하는 것이 어렵다**
  - LBA는 StegaStamp의 Decoder에 GradCAM을 사용해야 하는데, 이는 공격자가 StegaStamp의 Decoder에 대한 사전 지식이 바탕이 되어야 함을 의미한다
  - 즉, **공격자가 Decoder에 대한 접근성이 없을 경우 LBA는 구현될 수 없다**

Future Directions

1️⃣ StegaStamp를 가장 효과적으로 공격한 Regeneration Attack에 대한 연구 조사

- “IMAGE WATERMARKS ARE REMOVABLE USING CONTROLLABLE REGENERATION FROM CLEAN NOISE”
  - CtrlRegen이라는 새로운 공격 기법으로 StegaStamp, Tree-Ring을 효과적으로 제거함

2️⃣ 회전(Rotation) 공격에 대해 워터마크를 강인하게 만드는 방법에 대한 연구 조사

감사합니다:)