# 적대적 학습을 통한 다양한 안티포렌식에 강인한 딥페이크 탐지 모델 개발

한밭대학교
김지수, 김민지, 민지민, 장한얼

# 목차

## 서론

- 개발 배경 및 목적
- 핵심 해결 방법

## 본론

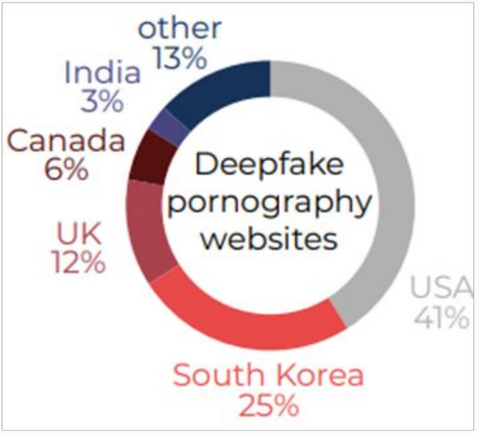
- 학습 네트워크 구조
- train 데이터셋 구축
- test 데이터셋 구축
- 모델 학습 기법

## 결론

- 실험 결과 및 분석

# 서론

**딥페이크는 사회적으로 악영향을 끼치는 기술**


**딥페이크 사용의 96% -> 포르노그라피**

**전세계 딥페이크 피해자 중 25% -> 한국인**

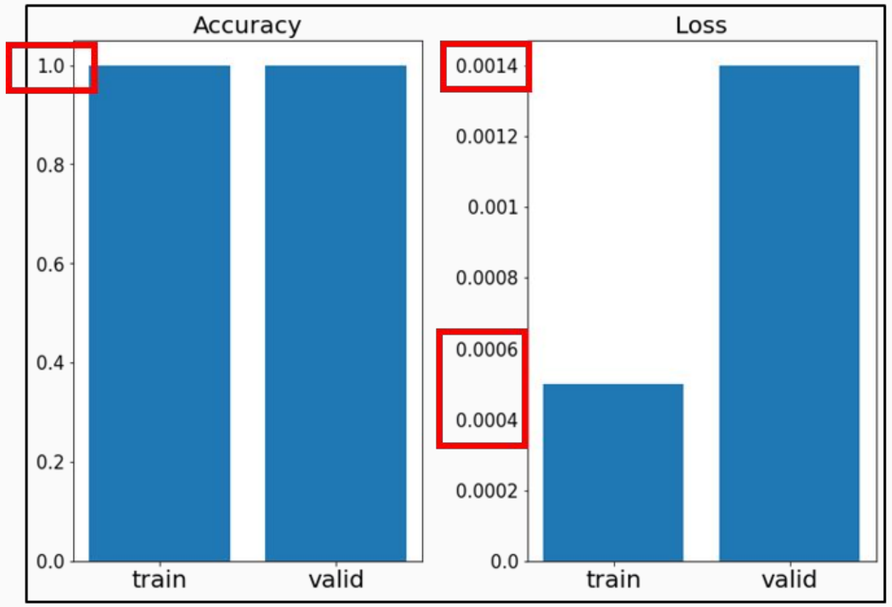
**딥페이크 탐지 모델에 대한 필요성 -> 연구 활발**

출처: The State of Deepfakes: Landscape, Threats, and Impact, Henry Ajder, Giorgio Patrini, Francesco Cavalli, and Laurence Cullen, September 2019.

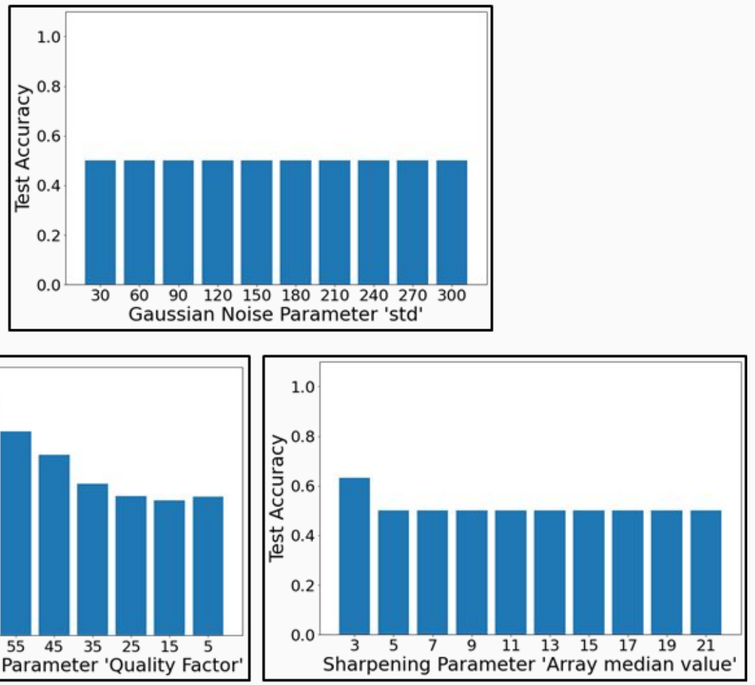
# 서론

## 딥페이크 탐지 모델의 무력화

- **간단한 이미지 편집**(ex. Sharpening, additive noise) 적용한 경우 **탐지 모델 -> 무력화**




원본 데이터셋 학습 모델 성능




편집된 이미지에 대한 정확도

# 서론

## 탐지 기술 무력화 -> 안티 포렌식

**적대적 데이터셋을 생성하는 안티 포렌식 공격**

- 화이트 박스 공격(White-Box Attacks)
  - 공격자가 **탐지 모델에 대한 정보를 모두 안다**는 전제
  - 비현실적 조건
  - 공격 성공률 100% 가까움

- 블랙 박스 공격(Black-Box Attacks)
  - 공격자가 **원본 이미지에 특정 노이즈 추가 -> 오분류 유도**
  - 실제 화이트 박스 공격보다 많이 시도됨

이미지 출처: https://sigport.org/documents/countering-anti-forensics-image-re-sampling#files

# 서론

## 두 가지의 블랙 박스 공격

- **Adversarial Attack**
  - 데이터셋에 네트워크를 교란시키는 노이즈를 추가
  - **초보자가 쉽게 공격할 수 있는 방법 X**

- **이미지 편집 기법**
  - 데이터셋에 이미지 편집을 가함
  - **초보자가 쉽게 공격할 수 있는 방법 O**

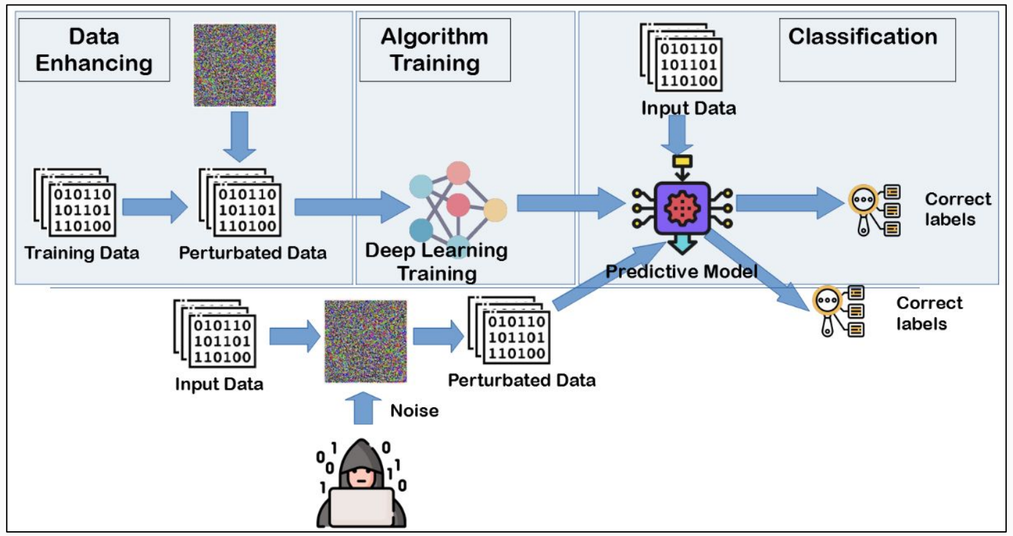
⇒ **두 방법 모두**
**딥페이크 탐지기 무력화 가능**

**전문가가 아니더라도 공격할 수 있는**
**이미지 편집 기법 기반의 블랙 박스 공격에 대응할 기술 필요**

# 본론

## 이미지 편집 기법을 이용한 블랙 박스 공격에 대응

⇒ 학습 데이터셋에 안티 포렌식 데이터셋 추가하는 **적대적 학습 수행**




**적대적 공격을 방어할 수 있도록**
**네트워크를 훈련시키는 방법**
→ 공격 패턴 학습하여 강인한 모델 생성

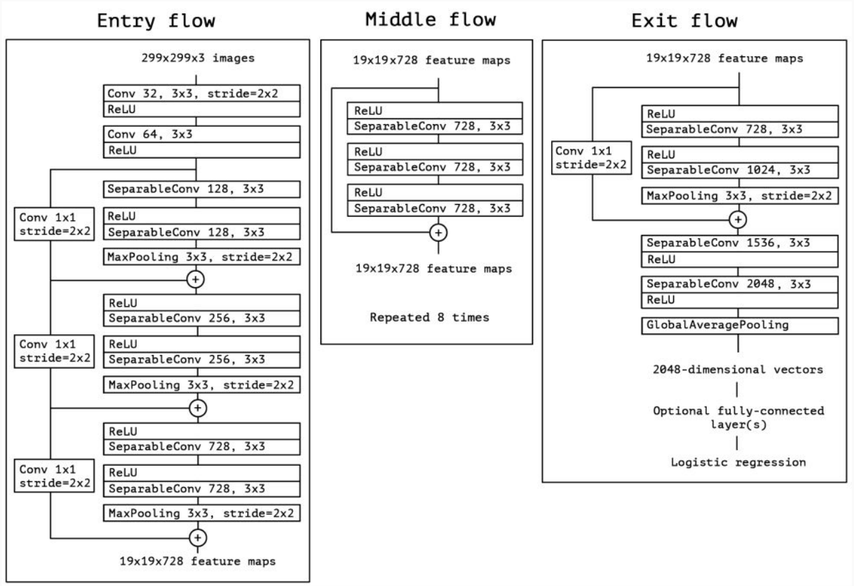
→ **다양한 유형의 안티포렌식 공격을 랜덤하게 적용**한 데이터셋을 적대적 학습에 사용
→ 강인한 모델 개발

이미지 출처: https://towardsdatascience.com/adversarial-machine-learning-mitigation-adversarial-learning-9ae04133c137

# 본론

## Xception 네트워크

→ 경험적으로 좋은 성능




네트워크 구조

[딥페이크 변조영상에 대한 탐지율]

| Accuracies | DF | F2F | FS | NT | Real | Total |
|---|---|---|---|---|---|---|
| Xcept. Full Image | 74.55 | 75.91 | 70.87 | 73.33 | 51.00 | 62.40 |
| Steg. Features | 73.64 | 73.72 | 68.93 | 63.33 | 34.00 | 51.80 |
| Cozzolino *et al.* | 85.45 | 67.88 | 73.79 | 78.00 | 34.40 | 55.20 |
| Rahmouni *et al.* | 85.45 | 64.23 | 56.31 | 60.07 | 50.00 | 58.10 |
| Bayar and Stamm | 84.55 | 73.72 | 82.52 | 70.67 | 46.20 | 61.60 |
| MesoNet | 87.27 | 56.20 | 61.17 | 40.67 | **72.60** | 66.00 |
| XceptionNet | **96.36** | **86.86** | **90.29** | **80.67** | 52.40 | **70.10** |

**출처:** Rossler, Andreas, et al. "**Faceforensics++**: Learning to detect manipulated facial images." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019.

네트워크 구조 이미지 출처:
https://openaccess.thecvf.com/content_cvpr_2017/html/Chollet_Xception_Deep_Learning_CVPR_2017_paper.html

# 본론

## 안티포렌식 데이터셋 구축

- Sharpening
- Median blur
- GoogLeNet Perturbation
- Gaussian Noise
- JPEG Compression

공격 순서
공격 강도
→ **RANDOM**

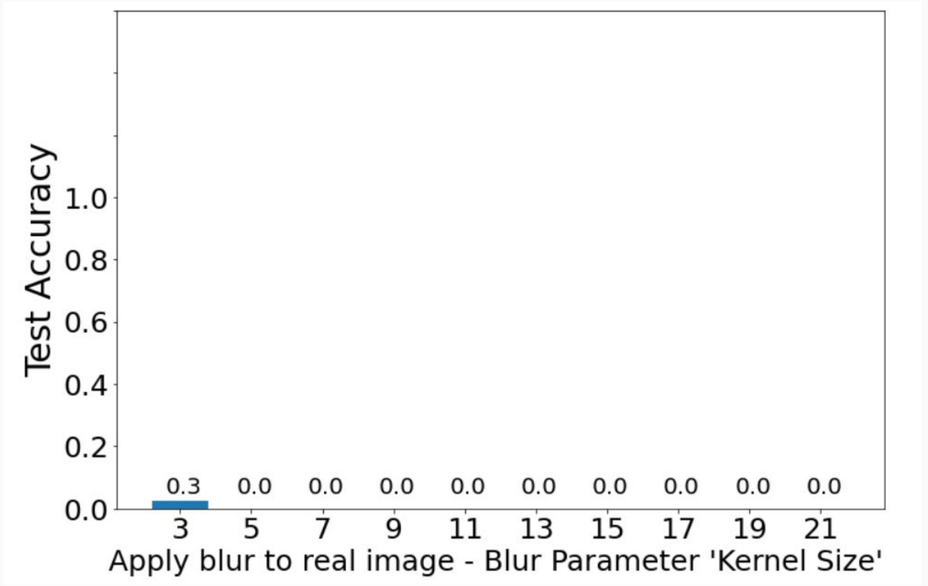
안티포렌식
데이터셋 생성

# 본론

## Median Blur를 사용한 이유

원본 fake 이미지

⇒ 딥페이크 변조 이미지는
합성된 부분에서 **blurry한 특징**




real 이미지에 median blur만 적용하여 실험해본 결과
⇒ 추가된 blur로 인해 **real 이미지를 fake로 오인**

**⇒ 따라서 blurry한 real 이미지가 fake 이미지로 판별되지 않도록 하기위해 Median Blur를 학습에 적용**

# 본론

## 안티포렌식 데이터셋

- train 데이터셋은 모든 이미지 편집 기법 적용
  → **랜덤한 파라미터 수치로 적용**

| Gaussian Noise | Median Blur | Sharpening | GoogleNet Perturbation | JPEG Compression |
|---|---|---|---|---|
| 30 ~ 210 | 3 ~ 15 | 0.5 ~ 3.5 | 0.05 ~ 0.35 | 90 ~ 35 |

- median blur는 noise를 제거할 수 있다고 판단하여 제일 첫순서로 적용
- JPEG Compression은 이미지의 최종 출력 형태가 JPEG 확장자이기 때문에 마지막으로 적용
- 남은 3가지 공격의 순서 → **랜덤**

# 본론

## 테스트 데이터셋 구축

다양한 공격에 강인한지 보기위해
→ 학습에 반영한 공격 + 학습에 반영하지 않은 공격

- 적대적 학습에 반영한 공격

  - Gaussian Noise
  - Median blur
  - Sharpening
  - GoogLeNet Perturbation
  - JPEG Compression

- 적대적 학습에 반영하지 않은 공격

  - Impulse Noise
  - Poisson Noise
  - Unsharp mask filter
  - ResNet Perturbation

# 본론

## 안티포렌식 test 데이터셋

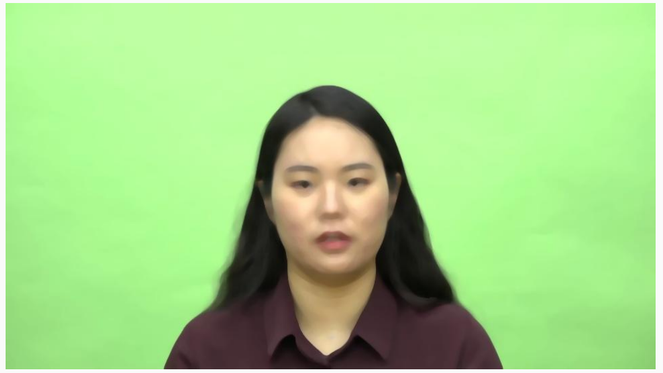
- 학습에 반영한 공격 파라미터

| Attack Level | Gaussian Noise | Median Blur | Sharpening | GoogLeNet Perturbation | JPEG Compression |
|---|---|---|---|---|---|
| 1 | 0.2 | 3 | 0.5 | 0.05 | 90 |
| 2 | 0.4 | 5 | 1.0 | 0.1 | 85 |
| 3 | 0.6 | 7 | 1.5 | 0.15 | 75 |
| 4 | 0.8 | 9 | 2.0 | 0.2 | 65 |
| 5 | 1 | 11 | 2.5 | 0.25 | 55 |
| 6 | 1.2 | 13 | 3.0 | 0.3 | 45 |
| 7 | 1.4 | 15 | 3.5 | 0.35 | 35 |
| 8 | 1.6 | 17 | 4.0 | 0.4 | 25 |
| 9 | 1.8 | 19 | 4.5 | 0.445 | 15 |
| 10 | 2 | 21 | 5.0 | 0.5 | 5 |

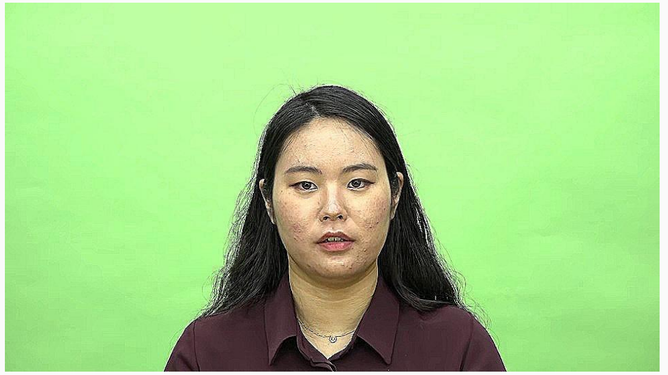
# 본론

## 안티포렌식 test 데이터셋

Gaussian Noise 예시 이미지

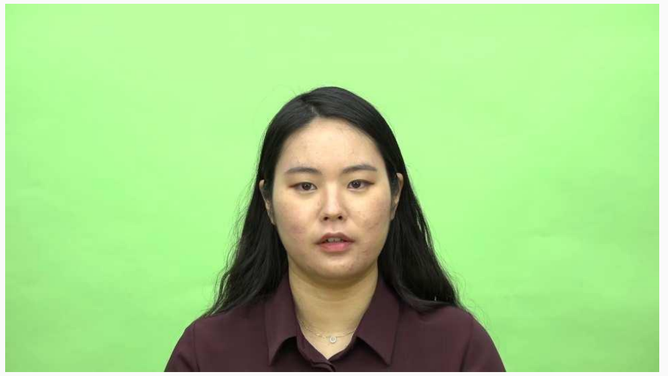


Median Blur 예시 이미지



Sharpening 예시 이미지

GoogLeNet Perturbation 예시 이미지



JPEG 예시 이미지

# 본론

## 안티포렌식 test 데이터셋

- 학습에 반영하지 않은 공격 파라미터

| Attack level | Poisson Noise | Impulse Noise | Unsharp mask filter | ResNet Perturbation |
|---|---|---|---|---|
| 1 | 0.2 | 1 | 0.05 | 0.05 |
| 2 | 0.4 | 3 | 0.1 | 0.1 |
| 3 | 0.6 | 5 | 0.15 | 0.15 |
| 4 | 0.8 | 6 | 0.2 | 0.2 |
| 5 | 1 | 7 | 0.25 | 0.25 |
| 6 | 1.2 | 9 | 0.3 | 0.3 |
| 7 | 1.4 | 11 | 0.35 | 0.35 |
| 8 | 1.6 | 13 | 0.4 | 0.4 |
| 9 | 1.8 | 15 | 0.45 | 0.45 |
| 10 | 2 | 17 | 0.5 | 0.2 |

# 본론

## 안티포렌식 test 데이터셋

Poisson Noise 예시 이미지

Impulse Noise 예시 이미지

Unsharp Mask 예시 이미지

ResNet Perturbation 예시 이미지

# 본론

## 안티포렌식 데이터셋 생성 시 적용한 사항

- **실제 발생할 수 있는 공격 상황은 예측 불가**
  → 공격(이미지 편집 기법) 별 강도 10단계 설정

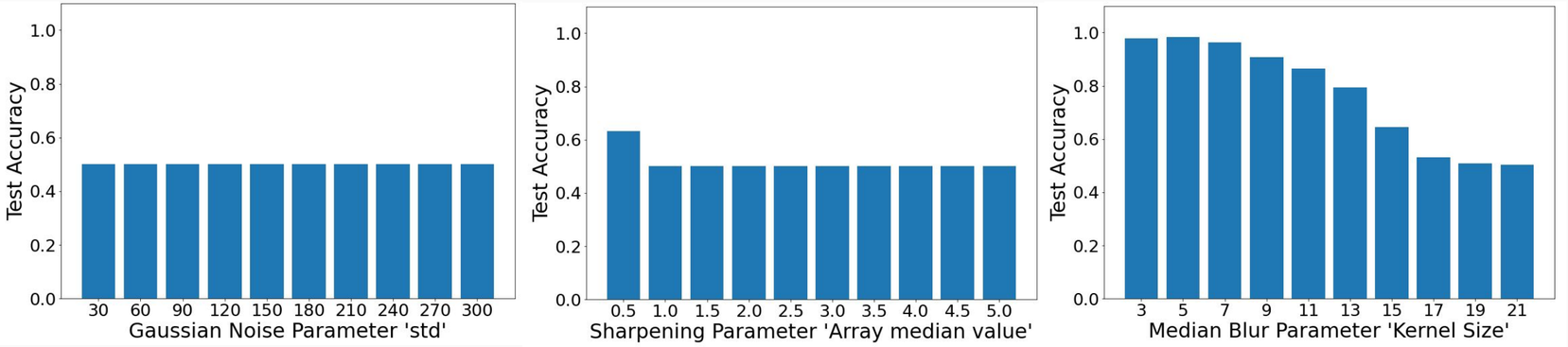
- **데이터셋에 적용할 공격 강도의 수준은 PSNR 사용해 조절**
  → PSNR 수치가 28이하인 이미지는 시각적 품질이 많이 떨어져 사용하지 않음

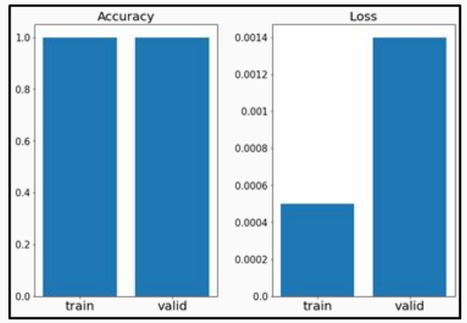
- **안티포렌식 공격의 목표 : real 이미지를 fake로 판별, fake 이미지를 real로 판별**
  → 데이터셋의 real 이미지와 fake 이미지에 모두 이미지 편집 적용

# 본론

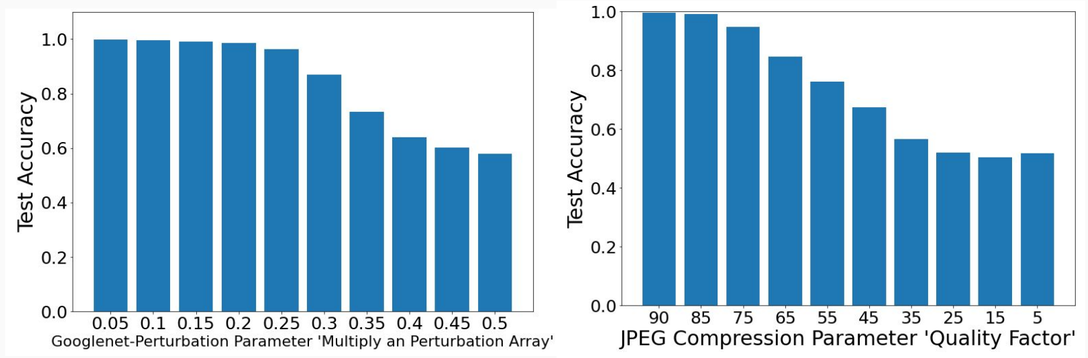
## 학습에 반영한 안티포렌식 데이터셋에 대한 성능 하락 확인

→ 원본 데이터셋만 학습한 모델 사용




원본 데이터셋 학습 모델 성능



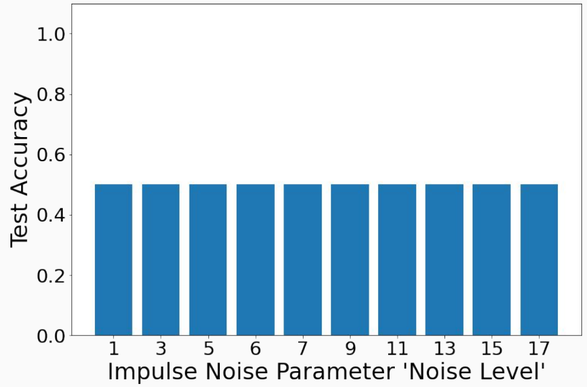

공격 강도 별 탐지 모델 **성능 매우 하락**

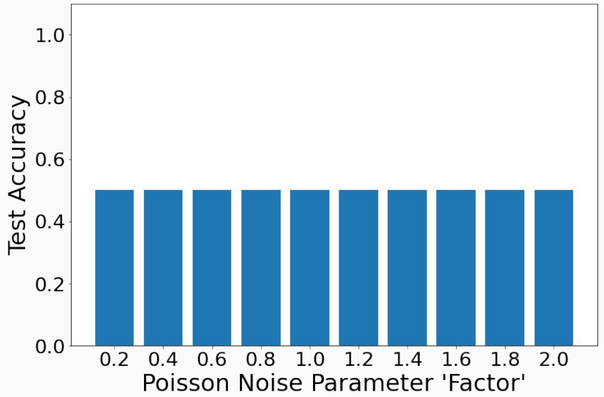
**=> 탐지 모델이 안티 포렌식에 취약**

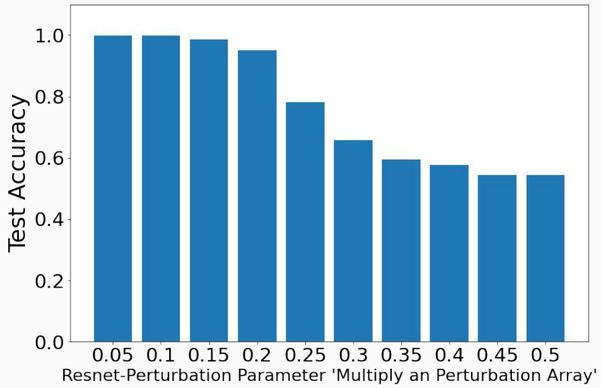
# 본론

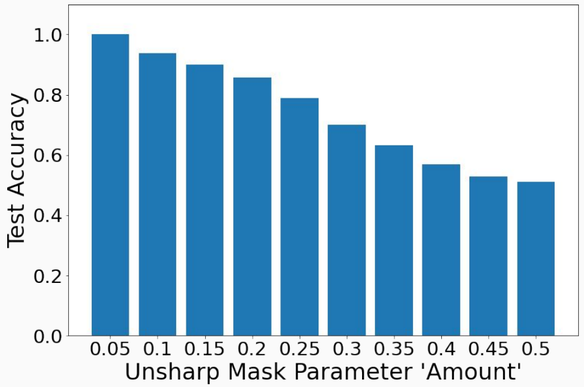
## 학습에 반영하지 않은 안티포렌식 데이터셋에 대한 성능 하락 확인

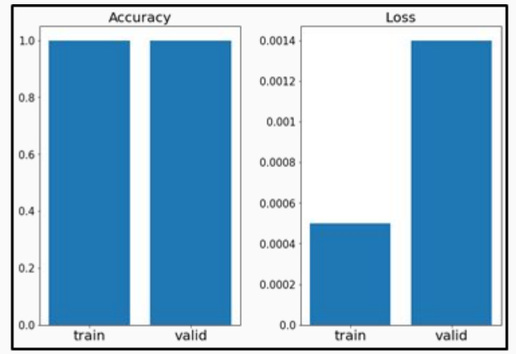
→ 원본 데이터셋만 학습한 모델 사용



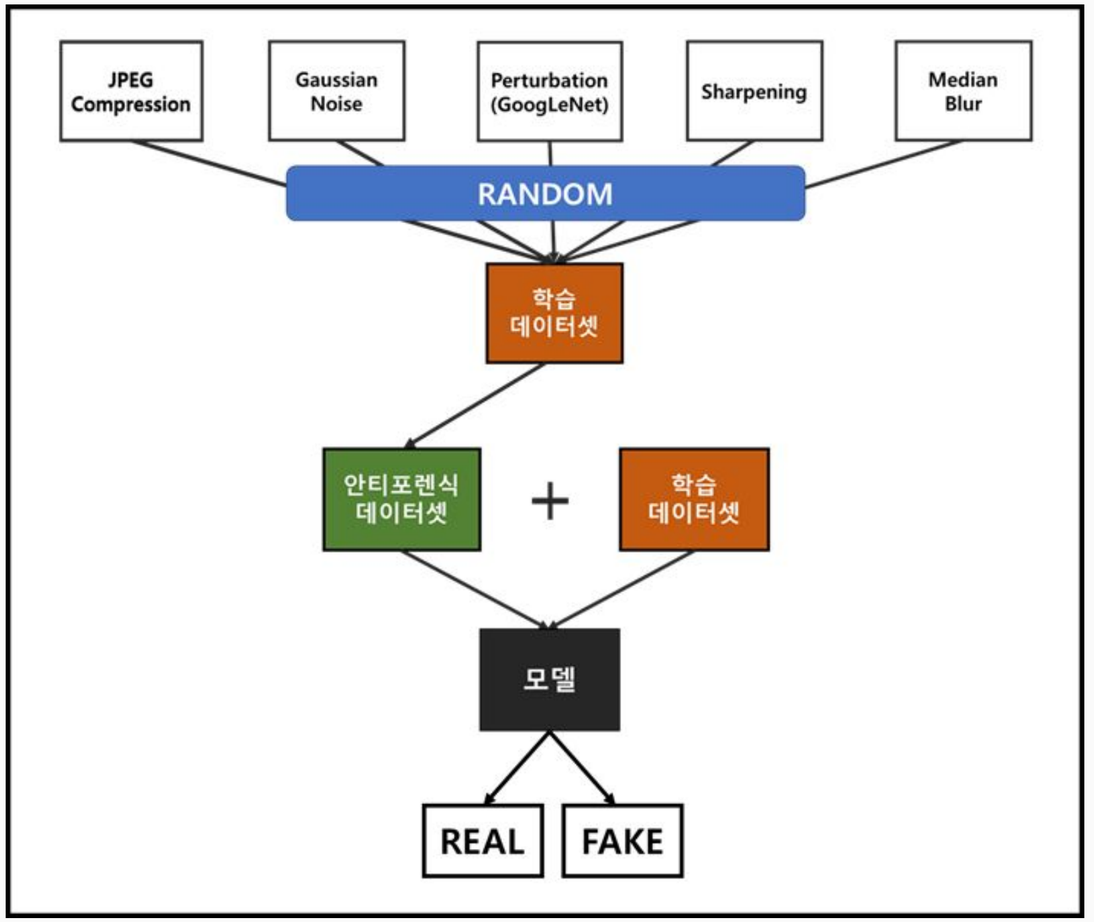

원본 데이터셋 학습 모델 성능

공격 강도 별 탐지 모델 **성능 매우 하락**

**=> 탐지 모델이 안티 포렌식에 취약**

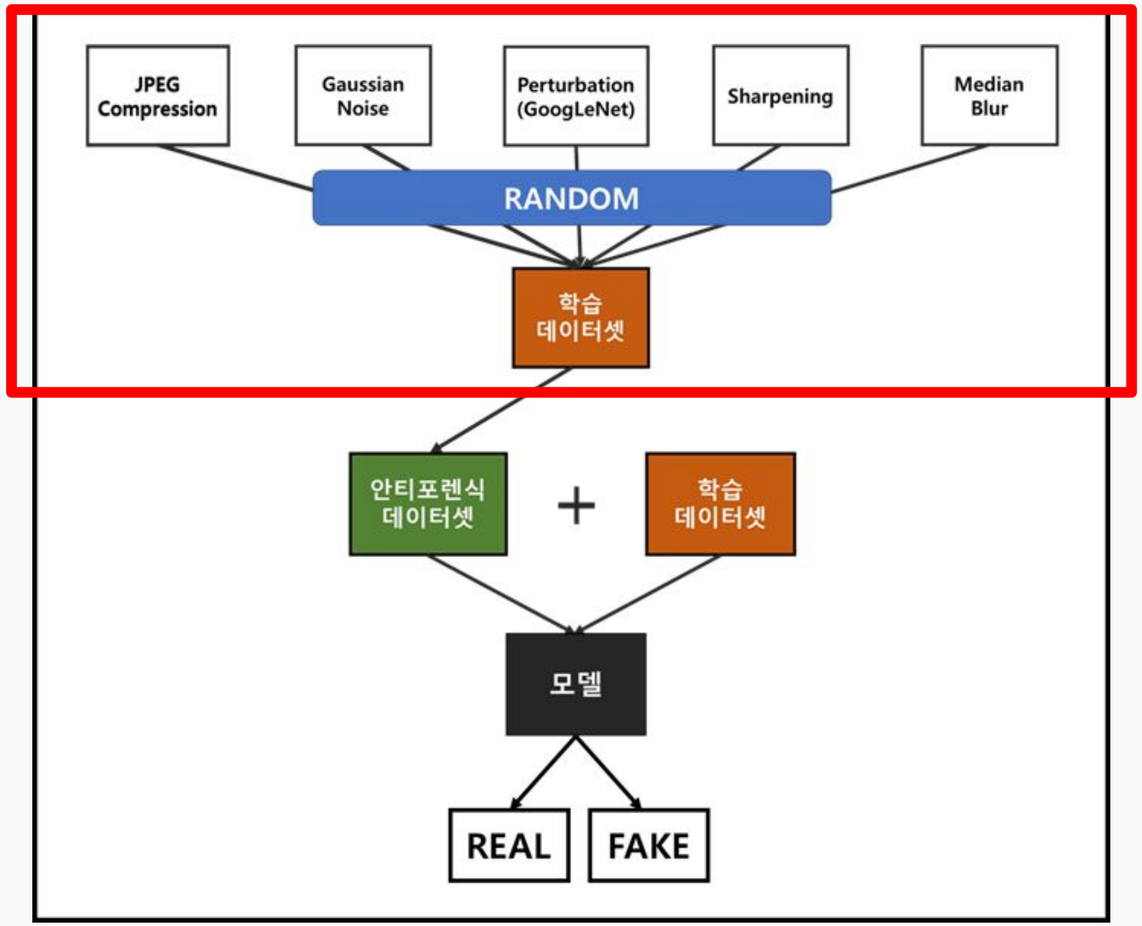
# 본론

## 제안하는 적대적 학습 기법



# 본론

## 제안하는 적대적 학습 기법 – 1단계

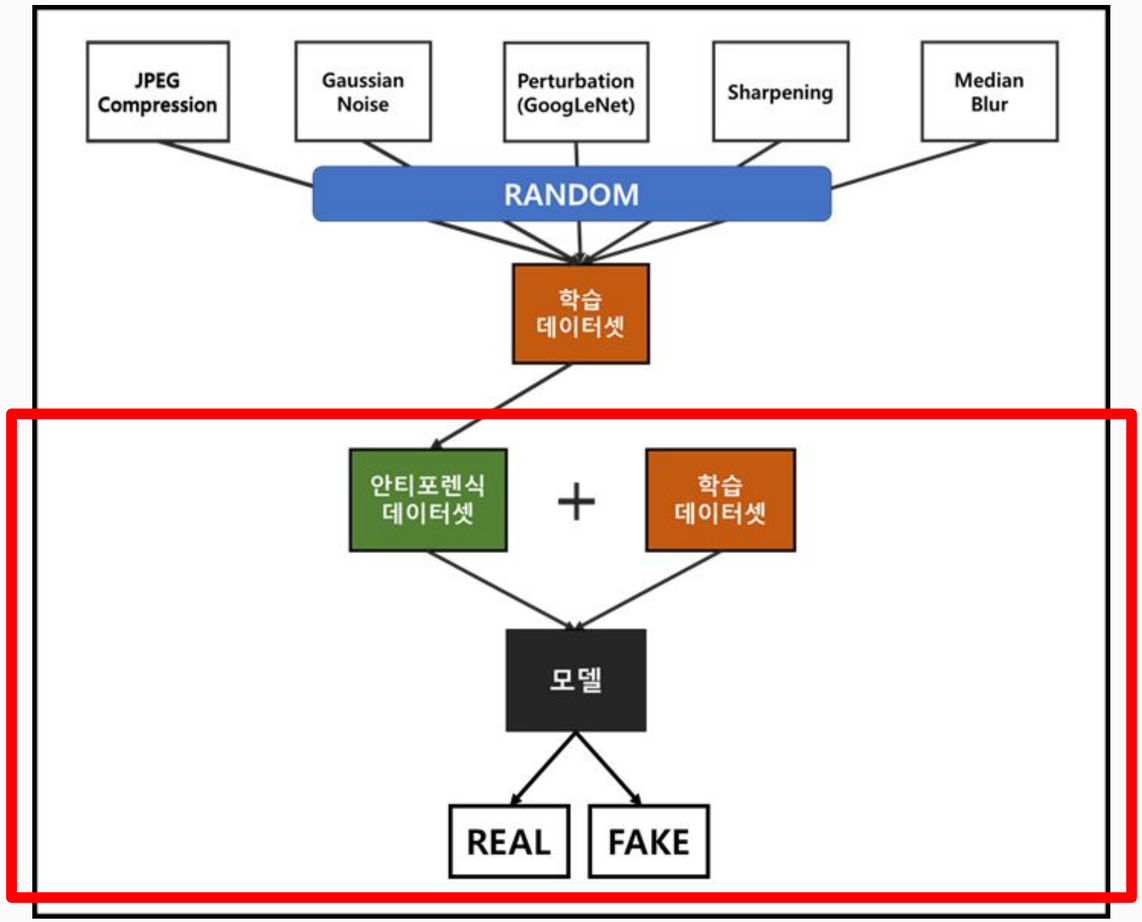



<안티포렌식 데이터셋 생성>

다양한 유형의 공격들을 random하게 적용한
안티포렌식 데이터셋 생성

# 본론

## 제안하는 적대적 학습 기법 – 2단계




<적대적 학습 수행>

원본 데이터셋과 안티포렌식 데이터셋을
학습 데이터셋으로 사용하는
적대적 학습으로 최종모델 도출

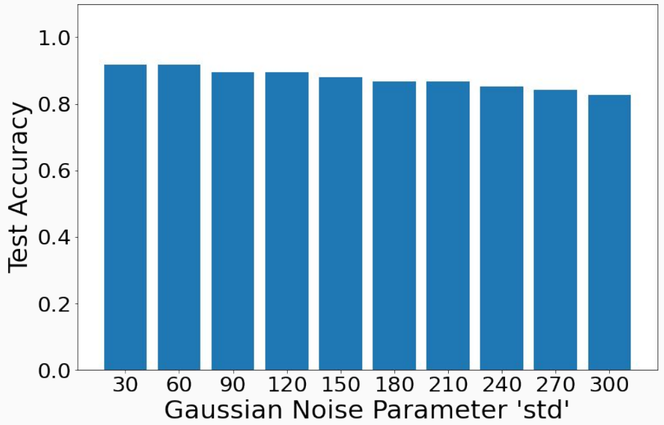
# 본론

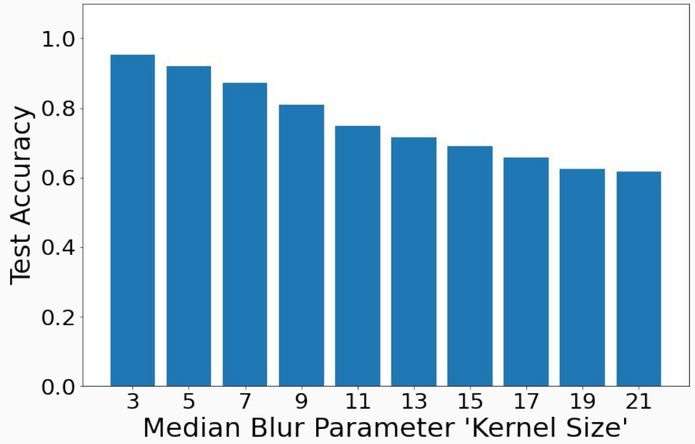
## 하이퍼파라미터 설정

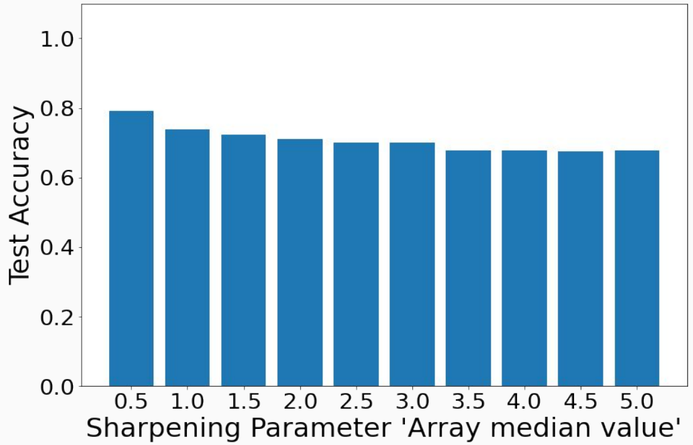
| | |
|---|---|
| **Epoch** | **10** |
| **Batch size** | 32 |
| **Optimizer** | Adam |
| **Learning rate** | 0.001 (초기값) |
| **LR Scheduler** | StepLR |
| **Loss** | CrossEntropyLoss |

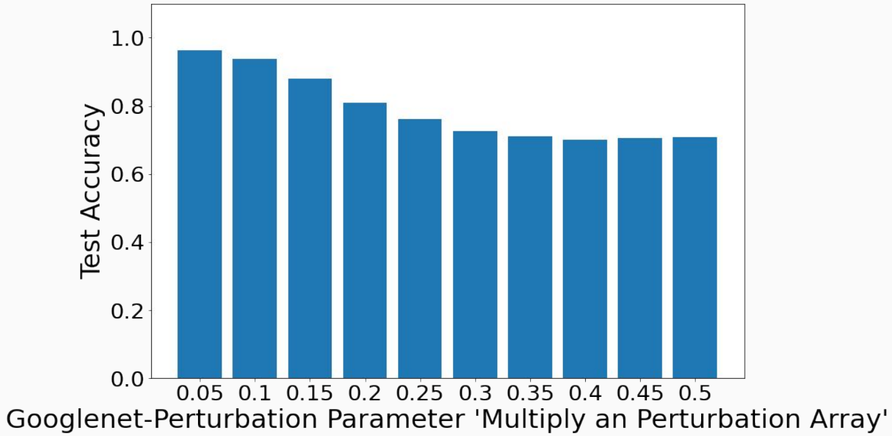
# 결론

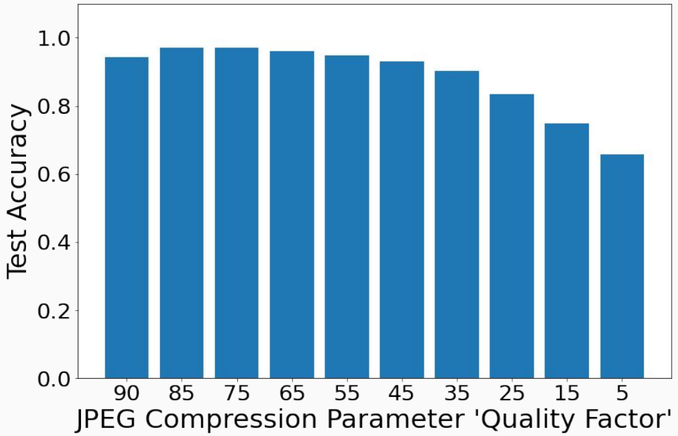
## 학습에 반영한 안티포렌식 데이터셋에 대한 성능



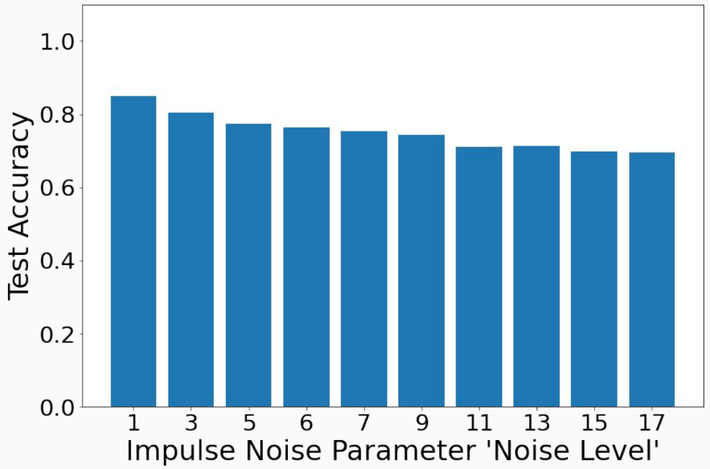



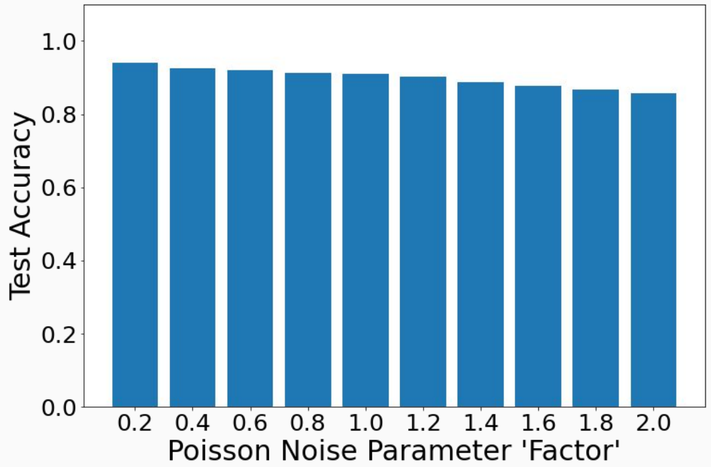



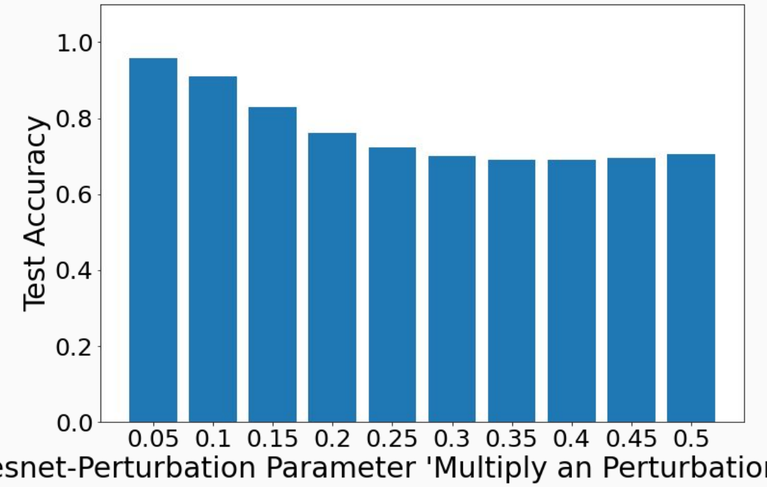

⇒ 적대적 학습 전과 비교하여
탐지율 상승확인

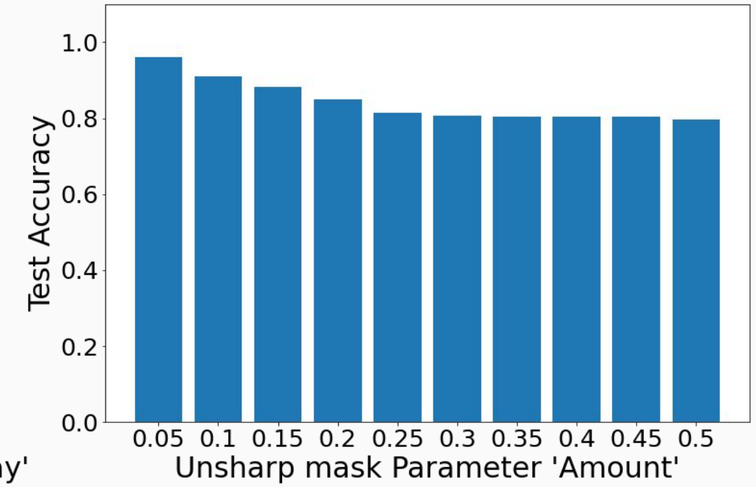
# 결론

## 학습에 반영하지 않은 안티포렌식 데이터셋에 대한 성능




⇒ 적대적 학습 전과 비교하여
탐지율 상승확인

# 결론

**기존 :** 탐지 정확도가 높은 모델이라도 **간단한 이미지 변형으로 탐지 우회** 가능

⬇

**적대적 학습 기법 적용** → 안티 포렌식 공격 패턴 학습

⬇

**학습에 반영한 공격들 뿐만아니라 학습에 반영하지 않은 공격들에 대해서도**
**높은 탐지 강인성 확인**

# Q&A