Increased-confidence Adversarial Examples for Deep Learning Counter-Forensics

Wenjie Li[1], Benedetta Tondi[2], Rongrong Ni[1], and Mauro Barni[2]

[1] Institute of Information Science, Beijing Jiaotong University, China

[2] Department of Information Engineering and Mathematics, University of Siena, Italy

Motivation

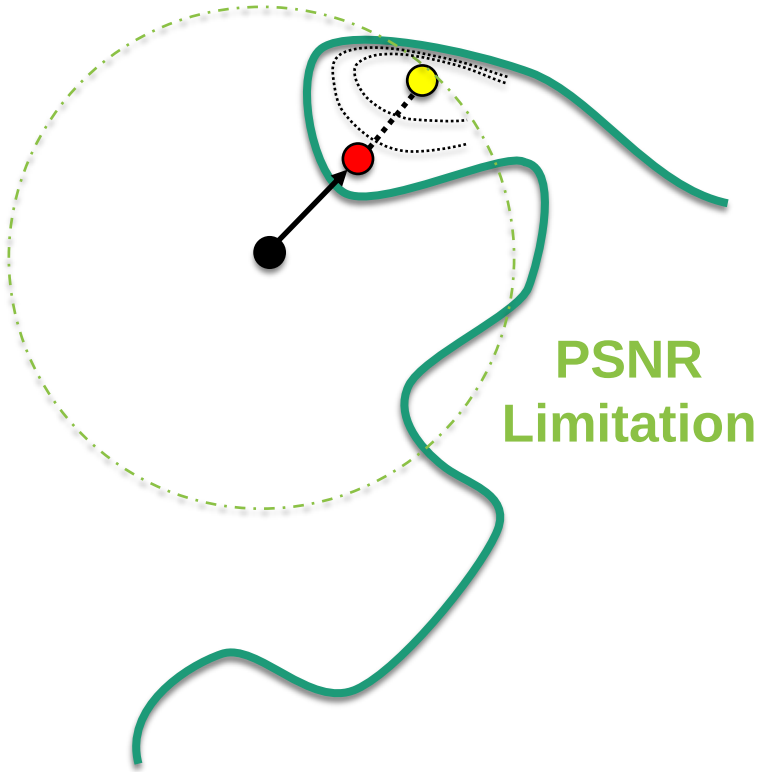
➢ Transferability of Attacks

- *Pattern Recognition*: adversarial examples against Deep Learning (DL) are often *transferable*
  - Powerful attacks can be carried out in *gray-box scenario*
- *Image Forensics*: adversarial examples are often *non-transferable*[1-3]
  - Common attack algorithms → minimize the distortion → the attack *fails* when the boundary is perturbed

[1] Barni M., Kallas K., Nowroozi E., Tondi B.: On the transferability of adversarial examples against CNN-based image forensics. ICASSP, 2019.
[2] Gragnaniello D., Marra F., Poggi G., Verdoliva L.: Analysis of adversarial attacks against CNN-based image forgery detectors. EUSIPCO, 2018.
[3] Marra F., Gragnaniello D., Verdoliva L.: On the vulnerability of deep learning to adversarial attacks for camera model identification. SPIC, 2018.

How to design stronger attacks ?

- Boundary of DL-based classifiers are often too complicated (especially for complicated tasks).
- Just increasing the distortion (e.g. PSNR limitation) is not a solution.



Proposed Confidence-controlled attacks

- We modified the *stop condition* of the attack in such a way to control the *confidence* of the misclassification.

Proposed Confidence-controlled attacks

- We modified the ***stop condition*** of the attack in such a way to control the ***confidence*** of the misclassification.
- Formalization (binary case)
  - X = input image, with class label $y = i$ (i=0,1); z_i = output logits (before softmax)

An image X' is judged to be adversarial *only if*

$$z_{1-i} - z_i > c,$$

where $c > 0$ is the desired minimum *confidence.*

Proposed Confidence-controlled attacks

- We modified the ***stop condition*** of the attack in such a way to control the ***confidence*** of the misclassification.
- Formalization (binary case)
  - X = input image, with class label $y = i$ (i=0,1); z_i = output logits (before softmax)

An image X' is judged to be adversarial *only if*

$$z_{1-i} - z_i > c,$$

where $c > 0$ is the desired minimum *confidence.*

- All the most common (iterative) attacks can be modified in this way.

Attack types

- (Base) Attack algorithms (gradient-based iterative attacks):
  - I-FGSM: iterative fast gradient sign method
  - PGD: I-FGSM with random projection of the starting point
  - MI-FGSM: momentum-based I-FGSM
  - C&W: an optimization-based method

Attack types

- Comparison:
  - DI^2-FGSM [1]: diverse input I-FGSM
    - A state-of-the-art method for more transferable adversarial examples
    - Diverse input - random transformations on the input image (random resizing and random padding)

[1] Xie C., Zhang Z., Zhou Y., Bai S., Wang J., Ren Z., Yuille A.L.: Improving transferability of adversarial examples with input diversity. CVPR, 2019.

Methodology

- Transferability assessment
  - Mismatch between source network (SN) and target network (TN)
    - **Cross-network** → different architectures, same dataset
    - **Cross-training** → same architecture, different datasets
    - **Cross-network-and-training** → different architectures, different datasets

Setup

- Detection tasks:
  - Median filtering (by a 5×5 window)
  - Image resizing (downsampling by 0.8)
  - Additive white Gaussian noise (AWGN, with std dev 1)

Setup

- Detection tasks:
  - Median filtering (by a 5×5 window)
  - Image resizing (downsampling by 0.8)
  - Additive white Gaussian noise (AWGN, with std dev 1)
- Datasets: RAISE and VISION

Setup

- Detection tasks:
  - Median filtering (by a 5×5 window)
  - Image resizing (downsampling by 0.8)
  - Additive white Gaussian noise (AWGN, with std dev 1)
- Datasets: RAISE and VISION
- Architectures: BSnet [1], BC+net [2], VGGnet [3]

[1] Bayar B., Stamm M.: A deep learning approach to universal image manipulation detection using a new convolutional layer. In: ACM Workshop on Info. Hiding & Multimedia Security. pp. 5-10, 2016
[2] Barni M., Costanzo A., Nowroozi E., Tondi B.: CNN-based detection of generic contrast adjustment with JPEG post-processing. ICIP, 2018.
[3] Simonyan K., Zisserman A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv: 1409.1556, 2014.

Experimental setting

- Model training and testing:
  - Training set: 2×10^5 for BSnet, 10^6 for BC+net, 10^5 for VGGnet
  - Testing set: 10^4 for BSnet, 5×10^4 for BC+net, 10^4 for VGGnet
  - Input size: 128×128

Experimental setting

- Model training and testing:
  - Training set: 2×10^5 for BSnet, 10^6 for BC+net, 10^5 for VGGnet
  - Testing set: 10^4 for BSnet, 5×10^4 for BC+net, 10^4 for VGGnet
  - Input size: 128×128
- Detection accuracy:
  - Median filtering: from 98.1% to 99.5%
  - Image resizing: from 96.6% to 99.0%
  - AWGN: from 98.3% to 99.9%

Experimental setting

- To-be-attacked: *500 manipulated images* from test set

Experimental setting

- To-be-attacked: *500 manipulated images* from test set
- The Foolbox [1] library is used to carry out the attacks
  - C&W
  - PGD: stepsize = 0.005, iterations = 100
  - I-FGSM: epsilons = 10, steps = 100
  - MI-FGSM: epsilons = 10, steps = 100, decay_factor = 0.2

[1] Rauber J., Brendel W., Bethge M.: Foolbox v0.8.0: A python toolbox to benchmark the robustness of machine learning models. 2017.

Experimental setting

- To-be-attacked: *500 manipulated images* from test set
- The Foolbox [1] library is used to carry out the attacks
  - C&W
  - PGD: stepsize = 0.005, iterations = 100
  - I-FGSM: epsilons = 10, steps = 100
  - MI-FGSM: epsilons = 10, steps = 100, decay_factor = 0.2
- DI^2-FGSM: the setting in [2] is followed
  - Resizing to $r \times r$ ($r \in [100,128)$) and random padding to 128×128

[1] Rauber J., Brendel W., Bethge M.: Foolbox v0.8.0: A python toolbox to benchmark the robustness of machine learning models. 2017.
[2] Xie C., Zhang Z., Zhou Y., Bai S., Wang J., Ren Z., Yuille A.L.: Improving transferability of adversarial examples with input diversity. CVPR, 2019.

Results - Cross-network (Median Filtering)

- SN = VGGnet on RAISE; TN = BSnet on RAISE

	C&W		PGD		I-FGSM		MI-FGSM		DI^2-FGSM			
c	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	iter	ASR_{SN}	ASR_{TN}	PSNR
0	8.4	69.1	5.4	67.5	27.2	58.1	27.2	58.1	1	86.4	51.8	48.3
12	50.6	54.6	55.0	52.0	71.0	47.9	71.6	47.8	2	97.0	69.2	45.0
12.5	70.1	51.5	79.0	48.9	88.2	45.3	88.6	45.3	3	99.2	77.6	43.2
13	91.2	48.1	94.6	45.4	96.4	42.5	96.6	42.7	5	100	86.8	41.4

- SN = BSnet on RAISE; TN = BC+net on RAISE

	C&W		PGD		I-FGSM		MI-FGSM		DI^2-FGSM			
c	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	iter	ASR_{SN}	ASR_{TN}	PSNR
0	0.2	72.0	0.2	74.5	23.2	59.7	23.4	59.7	1	91.4	52.4	48.2
50	56.0	52.2	55.6	50.3	60.6	48.9	60.4	48.9	2	98.4	69.4	44.6
80	74.0	47.8	74.0	45.8	77.8	45.1	78.8	44.8	3	99.4	77.0	42.3
100	83.6	45.2	83.6	43.5	85.4	42.9	86.2	42.7	5	100	85.4	40.2

Results - Cross-network (Median Filtering)

- SN = VGGnet on RAISE; TN = BSnet on RAISE

Attack success rate on TN

$ASR_{SN} \approx 100\%$

	C&W		PGD		I-FGSM		MI-FGSM		DI^2-FGSM			
c	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	iter	ASR_{SN}	ASR_{TN}	PSNR
0	8.4	69.1	5.4	67.5	27.2	58.1	27.2	58.1	1	86.4	51.8	48.3
12	50.6	54.6	55.0	52.0	71.0	47.9	71.6	47.8	2	97.0	69.2	45.0
12.5	70.1	51.5	79.0	48.9	88.2	45.3	88.6	45.3	3	99.2	77.6	43.2
13	91.2	48.1	94.6	45.4	96.4	42.5	96.6	42.7	5	100	86.8	41.4

- SN = BSnet on RAISE; TN = BC+net on RAISE

	C&W		PGD		I-FGSM		MI-FGSM		DI^2-FGSM			
c	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	iter	ASR_{SN}	ASR_{TN}	PSNR
0	0.2	72.0	0.2	74.5	23.2	59.7	23.4	59.7	1	91.4	52.4	48.2
50	56.0	52.2	55.6	50.3	60.6	48.9	60.4	48.9	2	98.4	69.4	44.6
80	74.0	47.8	74.0	45.8	77.8	45.1	78.8	44.8	3	99.4	77.0	42.3
100	83.6	45.2	83.6	43.5	85.4	42.9	86.2	42.7	5	100	85.4	40.2

Results - Cross-network (Median Filtering)

- SN = VGGnet on RAISE; TN = BSnet on RAISE

Attack success rate on TN

$ASR_{SN} \approx 100\%$

	C&W		PGD		I-FGSM		MI-FGSM		DI^2-FGSM			
c	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	iter	ASR_{SN}	ASR_{TN}	PSNR
0	8.4	69.1	5.4	67.5	27.2	58.1	27.2	58.1	1	86.4	51.8	48.3
12	50.6	54.6	55.0	52.0	71.0	47.9	71.6	47.8	2	97.0	69.2	45.0
12.5	70.1	51.5	79.0	48.9	88.2	45.3	88.6	45.3	3	99.2	77.6	43.2
13	91.2	48.1	94.6	45.4	96.4	42.5	96.6	42.7	5	100	86.8	41.4

- SN = BSnet on RAISE; TN = BC+net on RAISE

	C&W		PGD		I-FGSM		MI-FGSM		DI^2-FGSM			
c	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	iter	ASR_{SN}	ASR_{TN}	PSNR
0	0.2	72.0	0.2	74.5	23.2	59.7	23.4	59.7	1	91.4	52.4	48.2
50	56.0	52.2	55.6	50.3	60.6	48.9	60.4	48.9	2	98.4	69.4	44.6
80	74.0	47.8	74.0	45.8	77.8	45.1	78.8	44.8	3	99.4	77.0	42.3
100	83.6	45.2	83.6	43.5	85.4	42.9	86.2	42.7	5	100	85.4	40.2

Results - Cross-network (Median Filtering)

➢SN = VGGnet on RAISE; TN = BSnet on RAISE

Attack success rate on TN

$ASR_{SN} \approx 100\%$

	C&W		PGD		I-FGSM		MI-FGSM		DI^2-FGSM			
c	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	iter	ASR_{SN}	ASR_{TN}	PSNR
0	8.4	69.1	5.4	67.5	27.2	58.1	27.2	58.1	1	86.4	51.8	48.3
12	50.6	54.6	55.0	52.0	71.0	47.9	71.6	47.8	2	97.0	69.2	45.0
12.5	70.1	51.5	79.0	48.9	88.2	45.3	88.6	45.3	3	99.2	77.6	43.2
13	91.2	48.1	94.6	45.4	96.4	42.5	96.6	42.7	5	100	86.8	41.4

➢SN = BSnet on RAISE; TN = BC+net on RAISE

	C&W		PGD		I-FGSM		MI-FGSM		DI^2-FGSM			
c	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	iter	ASR_{SN}	ASR_{TN}	PSNR
0	0.2	72.0	0.2	74.5	23.2	59.7	23.4	59.7	1	91.4	52.4	48.2
50	56.0	52.2	55.6	50.3	60.6	48.9	60.4	48.9	2	98.4	69.4	44.6
80	74.0	47.8	74.0	45.8	77.8	45.1	78.8	44.8	3	99.4	77.0	42.3
100	83.6	45.2	83.6	43.5	85.4	42.9	86.2	42.7	5	100	85.4	40.2

Results - Cross-network (Median Filtering)

- SN = VGGnet on RAISE; TN = BSnet on RAISE

Attack success rate on TN

$ASR_{SN} \approx 100\%$

	C&W		PGD		I-FGSM		MI-FGSM		DI^2-FGSM			
c	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	iter	ASR_{SN}	ASR_{TN}	PSNR
0	8.4	69.1	5.4	67.5	27.2	58.1	27.2	58.1	1	86.4	51.8	48.3
12	50.6	54.6	55.0	52.0	71.0	47.9	71.6	47.8	2	97.0	69.2	45.0
12.5	70.1	51.5	79.0	48.9	88.2	45.3	88.6	45.3	3	99.2	77.6	43.2
13	91.2	48.1	94.6	45.4	96.4	42.5	96.6	42.7	5	100	86.8	41.4

- SN = BSnet on RAISE; TN = BC+net on RAISE

	C&W		PGD		I-FGSM		MI-FGSM		DI^2-FGSM			
c	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	iter	ASR_{SN}	ASR_{TN}	PSNR
0	0.2	72.0	0.2	74.5	23.2	59.7	23.4	59.7	1	91.4	52.4	48.2
50	56.0	52.2	55.6	50.3	60.6	48.9	60.4	48.9	2	98.4	69.4	44.6
80	74.0	47.8	74.0	45.8	77.8	45.1	78.8	44.8	3	99.4	77.0	42.3
100	83.6	45.2	83.6	43.5	85.4	42.9	86.2	42.7	5	100	85.4	40.2

Results - Cross-network (Resizing)

- SN = VGGnet on RAISE; TN = BSnet on RAISE

	C&W		PGD		I-FGSM		MI-FGSM		DI^2-FGSM			
c	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	iter	ASR_{SN}	ASR_{TN}	PSNR
0	1.2	71.5	1.8	75.4	0.4	59.3	0.4	59.2	1	30.6	2.6	48.2
17	40.4	36.8	22.0	33.4	25.0	33.3	24.8	33.3	25	100	8.6	32.6
18	53.6	34.3	39.8	30.9	39.6	30.9	37.4	30.9	35	100	23.4	30.3
19	64.4	32.2	52.0	28.9	51.6	28.9	52.6	28.9	45	100	41.0	28.2

- SN = BSnet on RAISE; TN = BC+net on RAISE

	C&W		PGD		I-FGSM		MI-FGSM		DI^2-FGSM			
c	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	iter	ASR_{SN}	ASR_{TN}	PSNR
0	0.4	68.3	0.4	66.9	0.4	58.7	0.4	58.6	1	96.8	0.2	48.2
50	82.4	45.9	65.2	42.3	66.2	41.9	63.6	41.7	3	100	33.8	41.2
80	85.2	39.3	84.0	35.5	82.4	35.3	84.6	35.4	8	100	77.0	35.3
100	80.4	34.0	82.8	31.6	83.8	31.6	82.2	31.7	15	100	87.6	31.1

Results - Cross-network (Resizing)

- SN = VGGnet on RAISE; TN = BSnet on RAISE

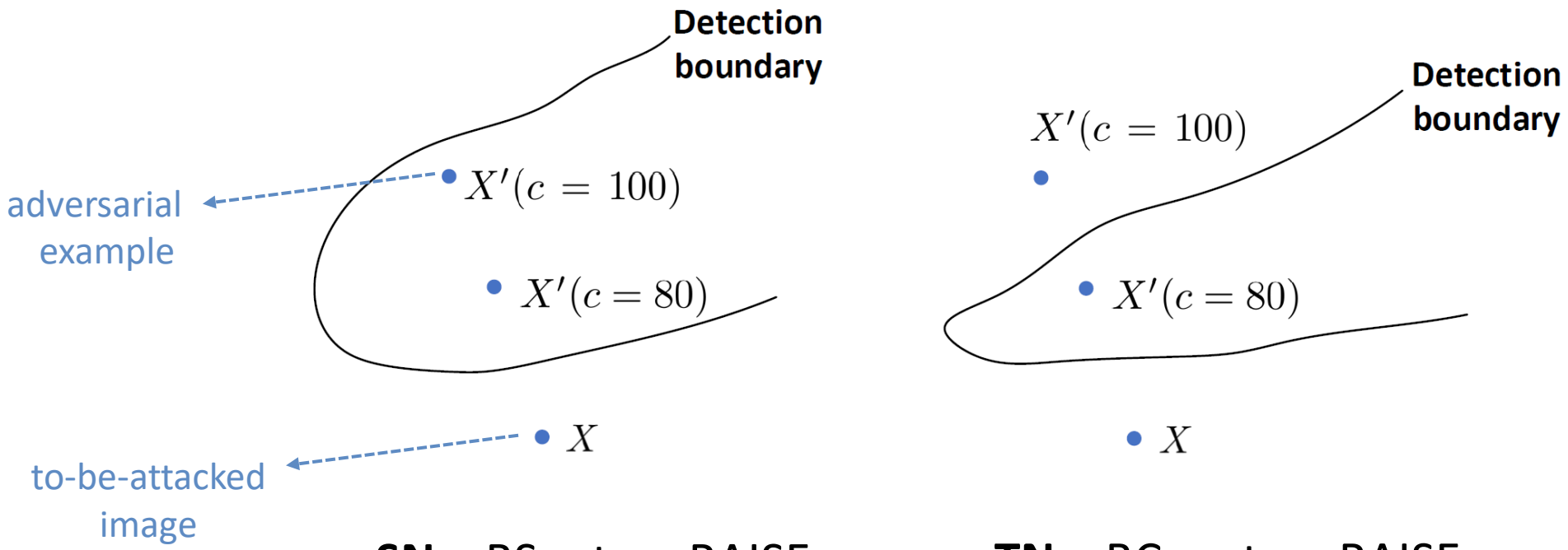
	C&W		PGD		I-FGSM		MI-FGSM		DI^2-FGSM			
c	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	iter	ASR_{SN}	ASR_{TN}	PSNR
0	1.2	71.5	1.8	75.4	0.4	59.3	0.4	59.2	1	30.6	2.6	48.2
17	40.4	36.8	22.0	33.4	25.0	33.3	24.8	33.3	25	100	8.6	32.6
18	53.6	34.3	39.8	30.9	39.6	30.9	37.4	30.9	35	100	23.4	30.3
19	64.4	32.2	52.0	28.9	51.6	28.9	52.6	28.9	45	100	41.0	28.2

- SN = BSnet on RAISE; TN = BC+net on RAISE

	C&W		PGD		I-FGSM		MI-FGSM		DI^2-FGSM			
c	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	iter	ASR_{SN}	ASR_{TN}	PSNR
0	0.4	68.3	0.4	66.9	0.4	58.7	0.4	58.6	1	96.8	0.2	48.2
50	82.4	45.9	65.2	42.3	66.2	41.9	63.6	41.7	3	100	33.8	41.2
80	85.2	39.3	84.0	35.5	82.4	35.3	84.6	35.4	8	100	77.0	35.3
100	80.4	34.0	82.8	31.6	83.8	31.6	82.2	31.7	15	100	87.6	31.1

A possible explanation

- Phenomenon: a larger confidence results in less transferability



SN = BSnet on RAISE

TN = BC+net on RAISE

Results – Cross-training

- Median Filtering: SN = BSnet on RAISE; TN = BSnet on VISION

	C&W		PGD		I-FGSM		MI-FGSM		DI^2-FGSM			
c	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	iter	ASR_{SN}	ASR_{TN}	PSNR
0	0.2	72.0	0.2	74.5	4.8	59.7	4.8	59.7	1	91.0	41.8	48.2
50	60.0	52.2	61.2	50.3	65.0	48.9	65.4	48.8	2	99.0	66.4	44.6
80	82.0	47.8	84.4	45.8	88.0	45.1	88.0	44.8	3	99.6	79.4	42.4
100	95.0	45.2	96.6	43.5	97.4	42.9	97.4	42.7	5	100	92.4	40.2

- Resizing: SN = BSnet on RAISE; TN = BSnet on VISION

	C&W		PGD		I-FGSM		MI-FGSM		DI^2-FGSM			
c	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	iter	ASR_{SN}	ASR_{TN}	PSNR
0	9.8	68.3	9.8	66.9	12.8	58.7	12.8	58.6	1	97.2	49.2	48.2
30	23.8	52.5	38.6	49.6	53.0	48.0	48.6	47.8	2	99.4	72.6	43.8
40	32.8	48.9	54.2	45.7	64.0	44.7	60.2	44.6	3	99.8	80.0	41.2
50	39.2	45.9	59.8	42.3	67.2	41.7	64.2	41.7	5	100	82.0	39.1

Results – Cross-training

- Median Filtering: SN = BSnet on RAISE; TN = BSnet on VISION

	C&W		PGD		I-FGSM		MI-FGSM		DI^2-FGSM			
c	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	iter	ASR_{SN}	ASR_{TN}	PSNR
0	0.2	72.0	0.2	74.5	4.8	59.7	4.8	59.7	1	91.0	41.8	48.2
50	60.0	52.2	61.2	50.3	65.0	48.9	65.4	48.8	2	99.0	66.4	44.6
80	82.0	47.8	84.4	45.8	88.0	45.1	88.0	44.8	3	99.6	79.4	42.4
100	95.0	45.2	96.6	43.5	97.4	42.9	97.4	42.7	5	100	92.4	40.2

- Resizing: SN = BSnet on RAISE; TN = BSnet on VISION

	C&W		PGD		I-FGSM		MI-FGSM		DI^2-FGSM			
c	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	iter	ASR_{SN}	ASR_{TN}	PSNR
0	9.8	68.3	9.8	66.9	12.8	58.7	12.8	58.6	1	97.2	49.2	48.2
30	23.8	52.5	38.6	49.6	53.0	48.0	48.6	47.8	2	99.4	72.6	43.8
40	32.8	48.9	54.2	45.7	64.0	44.7	60.2	44.6	3	99.8	80.0	41.2
50	39.2	45.9	59.8	42.3	67.2	41.7	64.2	41.7	5	100	82.0	39.1

Results – Cross-training

- Median Filtering: SN = BSnet on RAISE; TN = BSnet on VISION

	C&W		PGD		I-FGSM		MI-FGSM		DI^2-FGSM			
c	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	iter	ASR_{SN}	ASR_{TN}	PSNR
0	0.2	72.0	0.2	74.5	4.8	59.7	4.8	59.7	1	91.0	41.8	48.2
50	60.0	52.2	61.2	50.3	65.0	48.9	65.4	48.8	2	99.0	66.4	44.6
80	82.0	47.8	84.4	45.8	88.0	45.1	88.0	44.8	3	99.6	79.4	42.4
100	95.0	45.2	96.6	43.5	97.4	42.9	97.4	42.7	5	100	92.4	40.2

- Resizing: SN = BSnet on RAISE; TN = BSnet on VISION

	C&W		PGD		I-FGSM		MI-FGSM		DI^2-FGSM			
c	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	iter	ASR_{SN}	ASR_{TN}	PSNR
0	9.8	68.3	9.8	66.9	12.8	58.7	12.8	58.6	1	97.2	49.2	48.2
30	23.8	52.5	38.6	49.6	53.0	48.0	48.6	47.8	2	99.4	72.6	43.8
40	32.8	48.9	54.2	45.7	64.0	44.7	60.2	44.6	3	99.8	80.0	41.2
50	39.2	45.9	59.8	42.3	67.2	41.7	64.2	41.7	5	100	82.0	39.1

Results (AWGN detection)

- Cross-network: SN = VGGnet on RAISE; TN = BSnet on RAISE

	C&W		PGD		I-FGSM		MI-FGSM		DI^2-FGSM			
c	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	iter	ASR_{SN}	ASR_{TN}	PSNR
0	1.2	64.2	5.8	60.6	10.6	56.2	10.8	56.0	1	36.0	2.2	54.0
10	19.2	57.8	20.6	54.8	40.4	52.1	41.4	51.9	3	64.4	30.6	48.6
15	52.0	53.8	50.8	51.4	79.4	49.4	77.8	49.2	5	82.2	48.0	46.8
20	79.9	49.5	82.8	46.8	91.0	45.4	92.2	45.4	10	93.2	63.6	42.7

- Cross-training: SN = BSnet on RAISE; TN = BSnet on VISION

	C&W		PGD		I-FGSM		MI-FGSM		DI^2-FGSM			
c	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	iter	ASR_{SN}	ASR_{TN}	PSNR
0	0.2	65.0	0.2	62.4	0.6	57.7	0.6	57.6	1	51.2	0.6	54.1
20	12.2	54.4	11.0	52.2	13.8	49.7	15.0	49.6	10	80.0	4.8	43.9
30	78.0	49.4	77.2	47.3	54.4	44.8	72.8	45.1	20	91.6	10.2	40.0
40	95.4	45.5	94.0	43.0	88.2	41.0	93.0	41.3	30	98.4	21.4	37.4

Results (AWGN detection)

- Cross-network: SN = VGGnet on RAISE; TN = BSnet on RAISE

	C&W		PGD		I-FGSM		MI-FGSM		DI^2-FGSM			
c	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	iter	ASR_{SN}	ASR_{TN}	PSNR
0	1.2	64.2	5.8	60.6	10.6	56.2	10.8	56.0	1	36.0	2.2	54.0
10	19.2	57.8	20.6	54.8	40.4	52.1	41.4	51.9	3	64.4	30.6	48.6
15	52.0	53.8	50.8	51.4	79.4	49.4	77.8	49.2	5	82.2	48.0	46.8
20	79.9	49.5	82.8	46.8	91.0	45.4	92.2	45.4	10	93.2	63.6	42.7

- Cross-training: SN = BSnet on RAISE; TN = BSnet on VISION

	C&W		PGD		I-FGSM		MI-FGSM		DI^2-FGSM			
c	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	iter	ASR_{SN}	ASR_{TN}	PSNR
0	0.2	65.0	0.2	62.4	0.6	57.7	0.6	57.6	1	51.2	0.6	54.1
20	12.2	54.4	11.0	52.2	13.8	49.7	15.0	49.6	10	80.0	4.8	43.9
30	78.0	49.4	77.2	47.3	54.4	44.8	72.8	45.1	20	91.6	10.2	40.0
40	95.4	45.5	94.0	43.0	88.2	41.0	93.0	41.3	30	98.4	21.4	37.4

Results – Cross-network and training

- Median Filtering: SN = BSnet on VISION; TN = BC+net on RAISE

	C&W		PGD		I-FGSM		MI-FGSM		DI^2-FGSM			
c	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	iter	ASR_{SN}	ASR_{TN}	PSNR
0	0.0	70.5	0.0	71.9	2.6	60.0	2.6	60.0	1	95.4	35.6	48.2
100	78.0	45.1	83.0	43.1	84.6	42.4	85.6	42.4	5	100	82.0	40.5

Results – Cross-network and training

- Median Filtering: SN = BSnet on VISION; TN = BC+net on RAISE

	C&W		PGD		I-FGSM		MI-FGSM		DI^2-FGSM			
c	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	iter	ASR_{SN}	ASR_{TN}	PSNR
0	0.0	70.5	0.0	71.9	2.6	60.0	2.6	60.0	1	95.4	35.6	48.2
100	78.0	45.1	83.0	43.1	84.6	42.4	85.6	42.4	5	100	82.0	40.5

Results – Cross-network and training

- Median Filtering: SN = BSnet on VISION; TN = BC+net on RAISE

	C&W		PGD		I-FGSM		MI-FGSM		DI^2-FGSM			
c	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	ASR_{TN}	PSNR	iter	ASR_{SN}	ASR_{TN}	PSNR
0	0.0	70.5	0.0	71.9	2.6	60.0	2.6	60.0	1	95.4	35.6	48.2
100	78.0	45.1	83.0	43.1	84.6	42.4	85.6	42.4	5	100	82.0	40.5

Conclusions and future works

➢ Conclusions:

- A ***general strategy*** is proposed to control the strength of the attacks based on the ***confidence of the attack (logit level).***
- By increasing the confidence, the transferability can be improved while the PSNR remains good in most cases.

Conclusions and future works

➢ Conclusions:

- A ***general strategy*** is proposed to control the strength of the attacks based on the ***confidence of the attack (logit level).***
- By increasing the confidence, the transferability can be improved while the PSNR remains good in most cases.

➢ Future works:

- ***Use the proposed attack as benchmark to evaluate the security*** of existing defenses.
- Develop more ***powerful defense*** mechanisms.

Thanks for your attention!