Targeted Mismatch Adversarial Attack:
Query with a Flower to Retrieve the Tower

Tolias et al., ICCV 2019

Presented by: Woo Jae Kim

Table of Contents

- **Motivation**
- **Related Works**
- **Methods**
- **Experiments**
- **Limitations**

MOTIVATION

Problems of image search system

- **Nowadays, users' query information used in image search may not be protected**

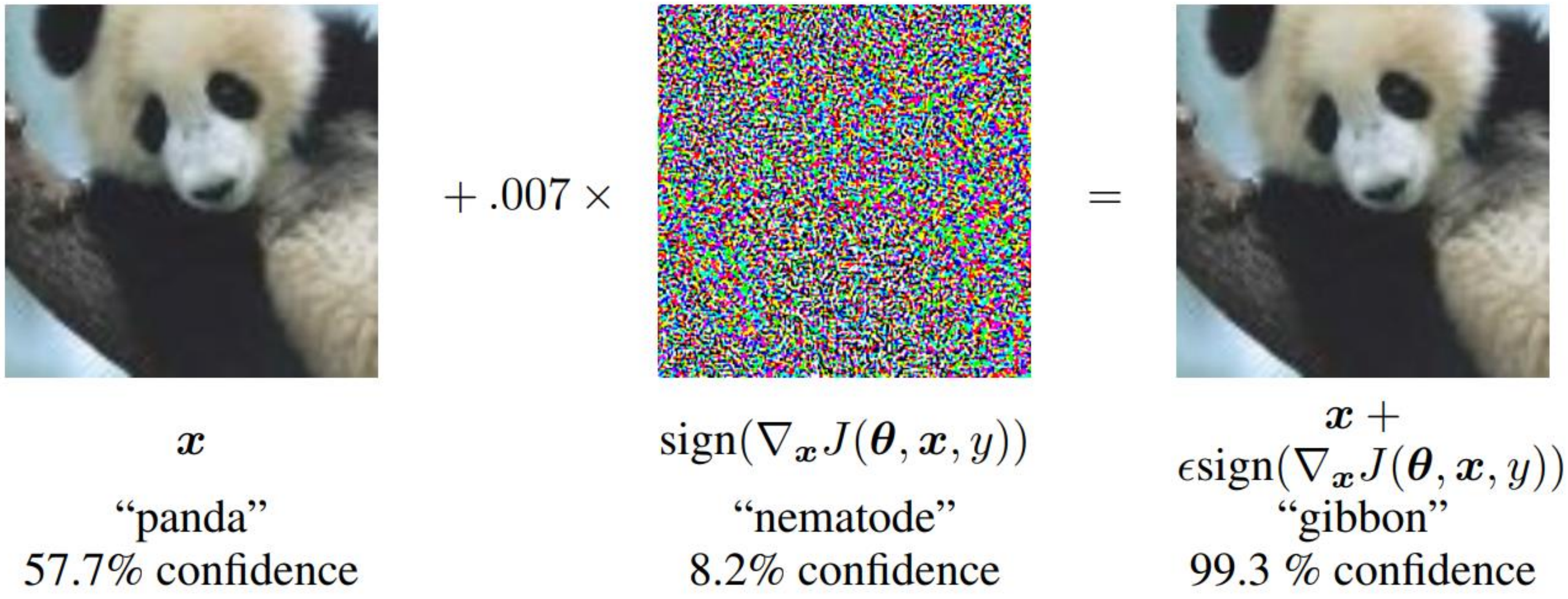
[1] Google Search Help: "The pictures you upload in your search may be stored by Google for 7 days. They won't be a part of your search history, and we'll only use them during that time to make our products and services better."

- **How can we protect our "personal" query?**

➔ Adversarial attack

What is adversarial attack?

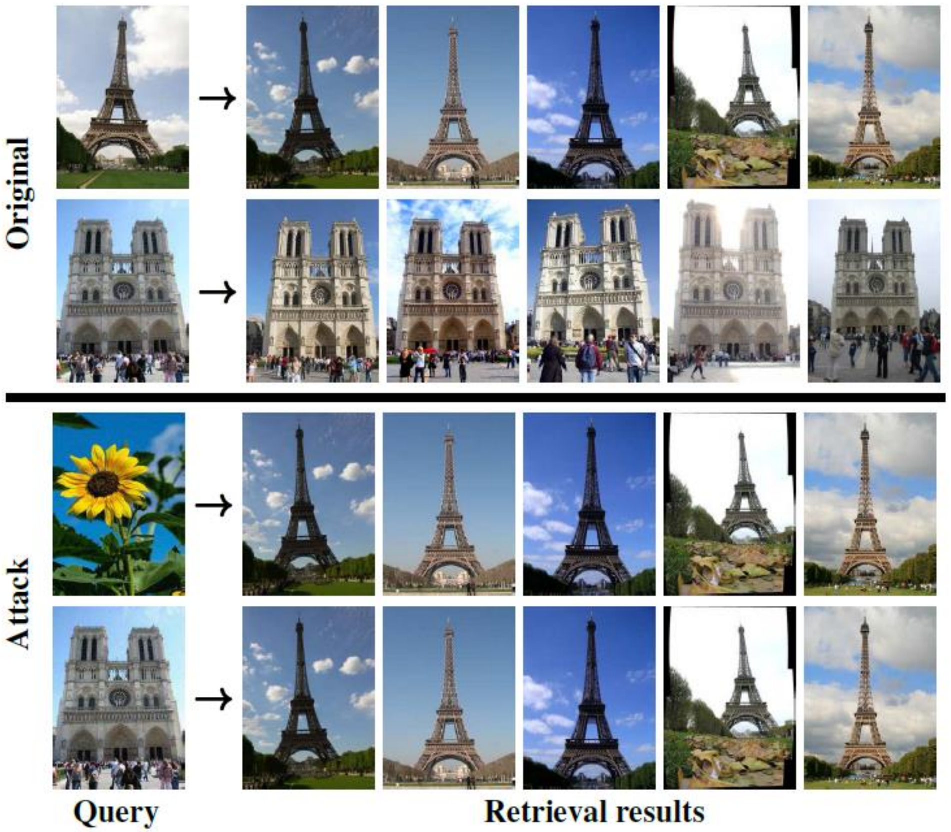
Adversarial attack: maliciously designed perturbation that when applied on image, causes a **machine learning model to make a mistake**



Explaining and Harnessing Adversarial Examples, ICLR 2015

How do we use adversarial attack?

- Aimed to fool DL-based image retrieval system
- Design adversarial query that return the **same search results as target query** but look **visually similar to carrier image**



RELATED WORKS

Adversarial attack

x = original image
y = gt label
x^{adv} = adversarial image
ϵ = perturbation scale
J_θ = classification loss of target classifier
$Clip_{x,\epsilon}$ = pixelwise clipping

- **Gradient-based attacks**
  - **Fast Gradient Sign Method (FGSM)**
    - **Maximizes first-order gradient of classification loss**

$$x^{adv} = x + \epsilon \, \mathrm{sign}\left(\nabla_x J_\theta(x, y)\right)$$

  - **Basic Iterative Method (BIM)**
    - **Iteratively repeats FGSM attack**

$$x_0^{adv} = x, \qquad x_{N+1}^{adv} = Clip_{x,\epsilon}\left\{x_N^{adv} + \alpha \, \mathrm{sign}\left(\nabla_x J_\theta\left(x_N^{adv}, y\right)\right)\right\}$$

Explaining and Harnessing Adversarial Examples, ICLR 2015
Adversarial Examples in the Physical World, ICLR Workshop 2017

Adversarial attack on image retrieval

- **Follows framework of adversarial attack on classification**
  - **Gradient-based approach**
  - **Generator-based approach**

- **However, these approaches used non-targeted attack**

- **Objective**

$$L_{\mathrm{nr}}(\mathbf{x}_c;\mathbf{x}) = \ell_{\mathrm{nr}}(\mathbf{x},\mathbf{x}_c) \quad +\lambda\,||\mathbf{x}-\mathbf{x}_c||^2$$
$$= \mathbf{h}_{\mathbf{x}}^\top \mathbf{h}_{\mathbf{x}_c} \quad +\lambda\,||\mathbf{x}-\mathbf{x}_c||^2$$

is optimized as:

$$\mathbf{x}_a = \arg\min_{\mathbf{x}} L_{\mathrm{nr}}(\mathbf{x}_c;\mathbf{x})$$

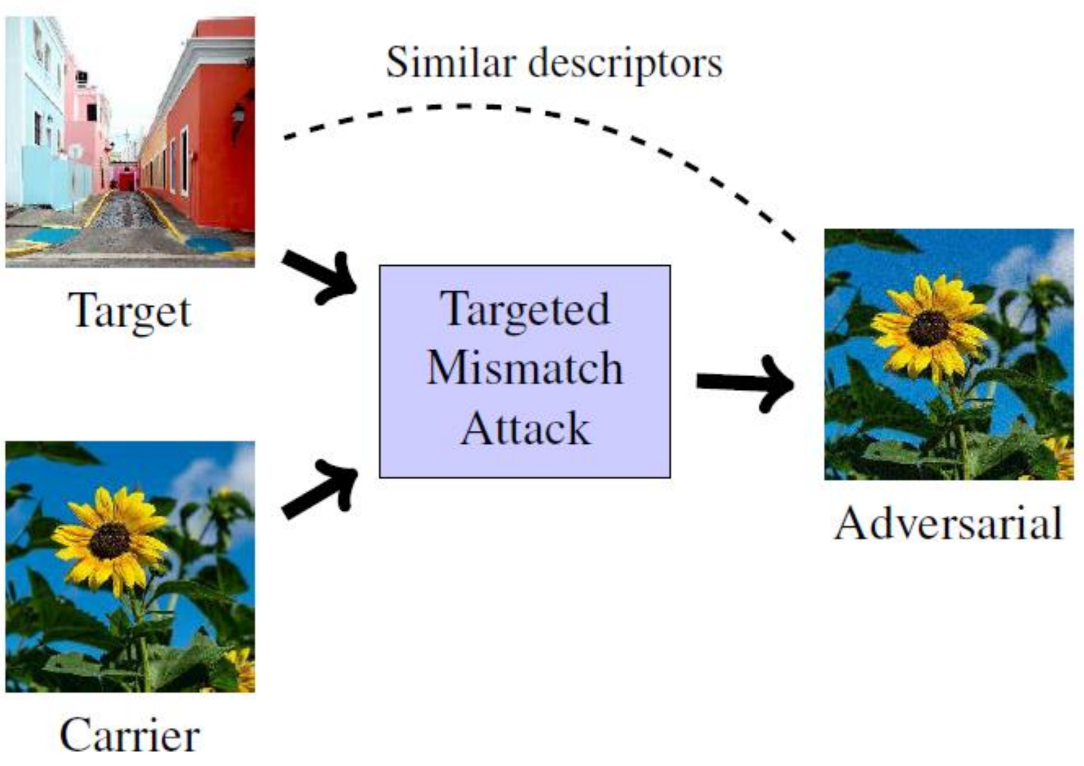
x = adversarial image
x_c = carrier image
y_c = gt label of carrier image
l_{nr} = performance loss

KAIST

METHODS

Problem formulation

- **Generate adversarial image that can be used to protect target query image**



KAIST

Problem formulation

- Generate **adversarial image** x that has high ***descriptor similarity*** but very low ***visual similarity*** to the target x_t

$$L_{\mathrm{tr}}(\mathbf{x}_c, \mathbf{x}_t; \mathbf{x}) = \ell_{\mathrm{tr}}(\mathbf{x}, \mathbf{x}_t) + \lambda \left|\left|\mathbf{x} - \mathbf{x}_c\right|\right|^2$$

Performance loss: make the descriptors of x similar to that of target image x_t

Distortion loss: make x visually similar to carrier image x_c

KAIST

Performance loss l_{tr}

x^s = image x with resolution s
g_x = feature descriptor of x
h_x = g_x passed through pooling layer
w_x = h_x passed through whitening
$u(g_x, b)_i$ = histogram of activations from the i-th channel of g_x on histogram bin centers b

- **Global descriptor**
  - **Suitable when all parameters of retrieval system are known**
  - **Can be l_{GeM}, l_{MAC}, etc … depending on pooling layer**

$$\ell_{\text{desc}}(\mathbf{x}, \mathbf{x}_t) = 1 - \mathbf{h}_{\mathbf{x}}^\top \mathbf{h}_{\mathbf{x}_t}$$

- **Activation tensor**
  - **Minimize the difference between features of x and x_t**

$$\ell_{\text{tens}}(\mathbf{x}, \mathbf{x}_t) = \frac{||\mathbf{g}_{\mathbf{x}} - \mathbf{g}_{\mathbf{x}_t}||^2}{w \cdot h \cdot d}$$

KAIST

x^s = image x with resolution s
g_x = feature descriptor of x
h_x = g_x passed through pooling layer
w_x = h_x passed through whitening
$u(g_x, b)_i$ = histogram of activations from the i-th channel of g_x on histogram bin centers b

Performance loss l_{tr}

- **Activation histogram**
  - **Minimize distance on first-order statistics of feature g_x**

$$\ell_{\text{hist}}(\mathbf{x}, \mathbf{x}_t) = \frac{1}{d} \sum_{i=1}^{d} ||u(\mathbf{g}_\mathbf{x}, \mathbf{b})_i - u(\mathbf{g}_{\mathbf{x}_t}, \mathbf{b})_i||$$

- **Different image resolution**
  - **Ensures that attack is successful across different resolutions**
  - **Often applies Gaussian blur on x^s to generate $x^{\hat{s}}$**

$$L^s_{\text{tr}}(\mathbf{x}, \mathbf{x}_t; \mathbf{x}) = \ell_{\text{tr}}(\mathbf{x}^s, \mathbf{x}^s_t) + \lambda \, ||\mathbf{x} - \mathbf{x}_c||^2$$

KAIST

Performance loss l_{tr}

x^s = image x with resolution s
g_x = feature descriptor of x
h_x = g_x passed through pooling layer
w_x = h_x passed through whitening
$u(g_x, b)_i$ = histogram of activations from the i-th channel of g_x on histogram bin centers b

- **Ensemble**
  - **Combine l_{desc} for all possible pooling layers $\mathcal{P}$**

$$\ell_{\mathcal{P}}(\mathbf{x}, \mathbf{x}_t) = \frac{\sum_{p \in \mathcal{P}} \ell_p(\mathbf{x}, \mathbf{x}_t)}{|\mathcal{P}|}$$

KAIST

Optimization

- **Adversarial image is generated by minimizing L_{tr}**
- **Uses gradient-based methods**

$$L_{\text{tr}}(\mathbf{x}_c, \mathbf{x}_t; \mathbf{x}) = \ell_{\text{tr}}(\mathbf{x}, \mathbf{x}_t) + \lambda \left|\left|\mathbf{x} - \mathbf{x}_c\right|\right|^2$$

$$\mathbf{x}_a = \arg\min_{\mathbf{x}} L_{\text{tr}}(\mathbf{x}_c, \mathbf{x}_t; \mathbf{x})$$

EXPERIMENTS

Experimental setup

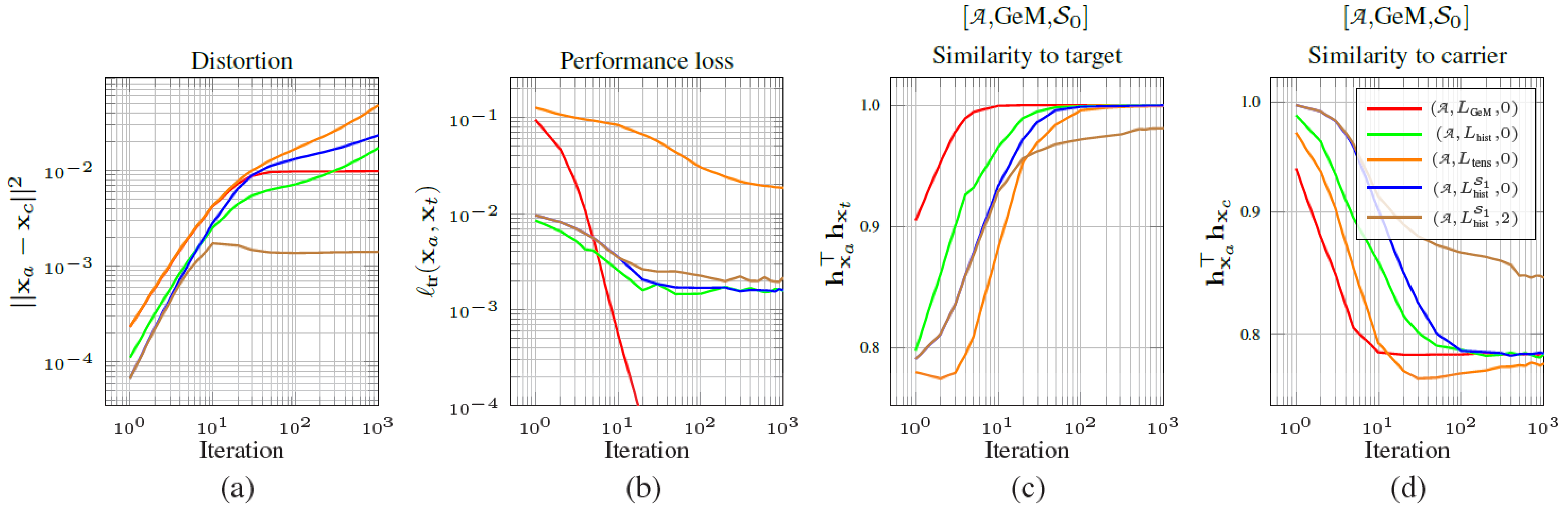
- **Datasets**
  - **Holidays, Copydays, $\mathcal{R}$Oxford, $\mathcal{R}$Paris**
- **Learning rate = 0.01, # iterations = 100 or 1000 (for L_{tens})**
- **Resolutions =**

$$^4\mathcal{S}_0 = \{1024\},\ \mathcal{S}_1 = \mathcal{S}_0 \cup \{300, 400, 500, 600, 700, 800, 900\},\ \mathcal{S}_2 = \mathcal{S}_1 \cup \{350, 450, 550, 650, 750, 850, 950\},\ \mathcal{S}_3 = \mathcal{S}_0 \cup \{262, 289, 319, 351, 387, 427, 470, 518, 571, 630, 694, 765, 843, 929\}$$

- **Target models**
  - **AlexNet ($\mathcal{A}$), ResNet18 ($\mathcal{R}$), VGG16 ($\mathcal{V}$)**
- $(\mathcal{A}, L_{hist}^{S1}, 0)$ – **optimization** on **AlexNet** using L_{hist}^{S1} with $\lambda = 0$
- $[\mathcal{A}, \mathrm{GeM}, S_0]$ – **testing** on **AlexNet** with test-pooling **GeM** and resolution S_0

Optimization iterations

- **Carrier distortion – increases as # iterations increases**
- **Performance loss (l_{tr}) – decreases as # iterations increases**
- **Similarity to target/carrier – increases/decreases as # iterations increases**



Robustness to unknown test-pooling

- **Mean average precision (mAP) and similarity ($x_t^\top x_a$) on different performance loss**
- **Adversarial query is tested under multiple test-pooling layers**

h \ L_{tr}	Original	L_{GeM}	$L_{\mathcal{P}}$	L_{hist}	L_{tens}
	mAP	mAP difference to original			
GeM	41.3	−0.0	−0.0	−0.2	−0.1
MAC	37.0	−0.5	−0.0	−0.8	−0.0
SPoC	32.9	−4.4	−0.1	−0.1	−0.7
R-MAC	44.1	−1.2	−0.5	−0.7	−0.0
CroW	38.2	−1.3	−0.4	−0.2	−0.0
	$\mathbf{x}_t^\top \mathbf{x}_a$				
GeM	1.000	1.000	1.000	0.997	0.998
MAC	1.000	0.972	1.000	0.985	0.996
SPoC	1.000	0.909	1.000	0.999	0.996
R-MAC	1.000	0.972	0.978	0.979	0.997
CroW	1.000	0.968	0.994	0.995	0.998

KAIST

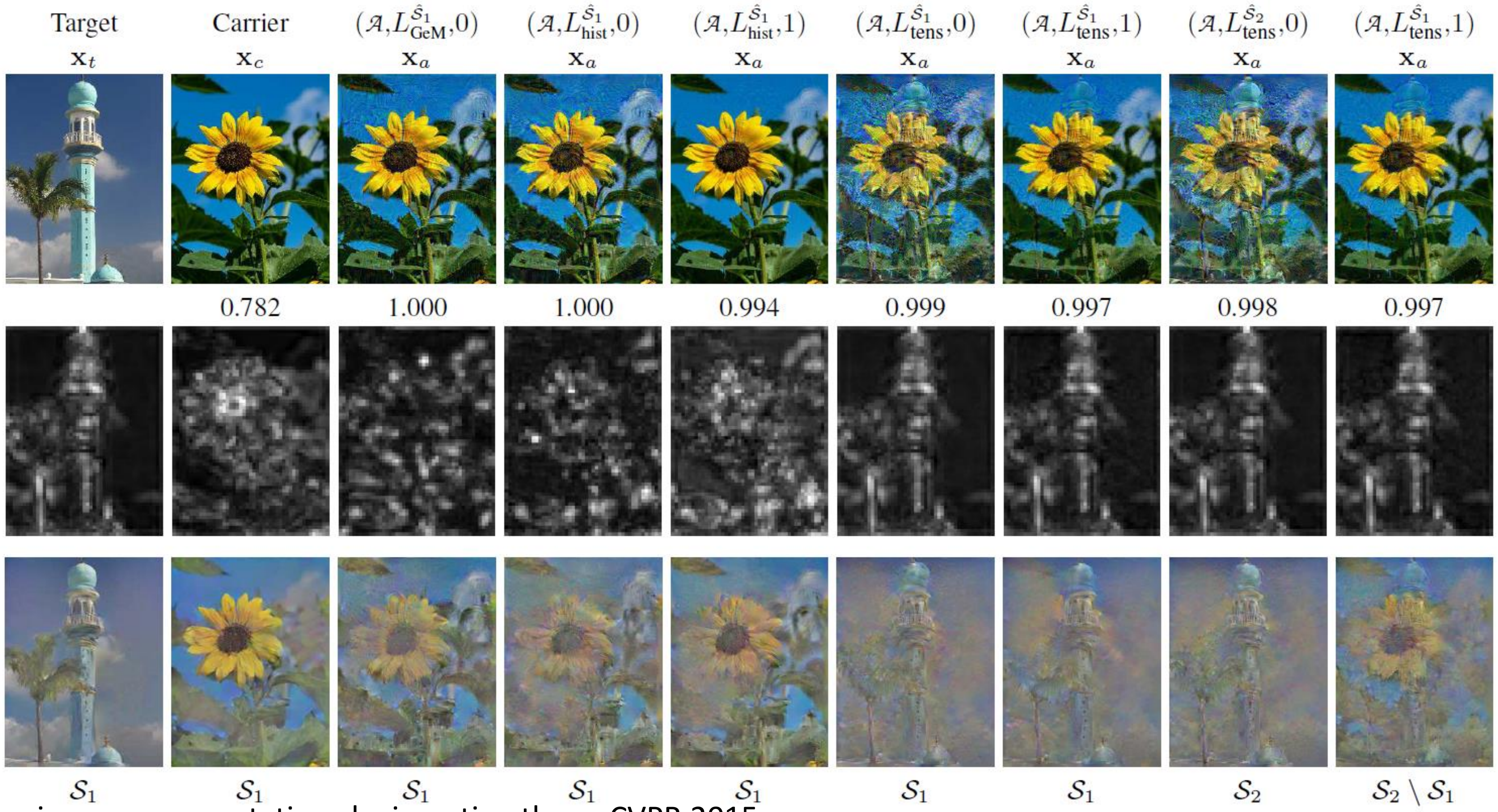
Impact of distortion term

- **Impact of λ on visualization of adversarial image**
- **Numbers below each image represent descriptor similarity with x_t**

KAIST

Concealing/revealing the target

- **Target, carrier, and adv. images (top row), depth-wise maximum of g (middle row), and inversion[1] of g (bottom row)**



1. Understanding deep image representations by inverting them, CVPR 2015

LIMITATIONS

Personal reflections

- **The usage of distortion loss $\| x - x_c \|^2$ is poorly justified**
  - **Even when its weight is 0, adversarial image retains high visual similarity to the carrier image**

- **Time taken for attack is too high**
  - **Optimization takes up to *68.4 sec* on certain cases**
  - **Not practical on large-scale search with high # of queries**

- **Paper lacks experiments/analysis on black-box models**
  - **Practically, the models used for retrieval tasks are unknown**
  - **Proposed method may show limited performance when the model is not known**

KAIST