JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

Enhancing the Self-Universality for Transferable Targeted Attacks

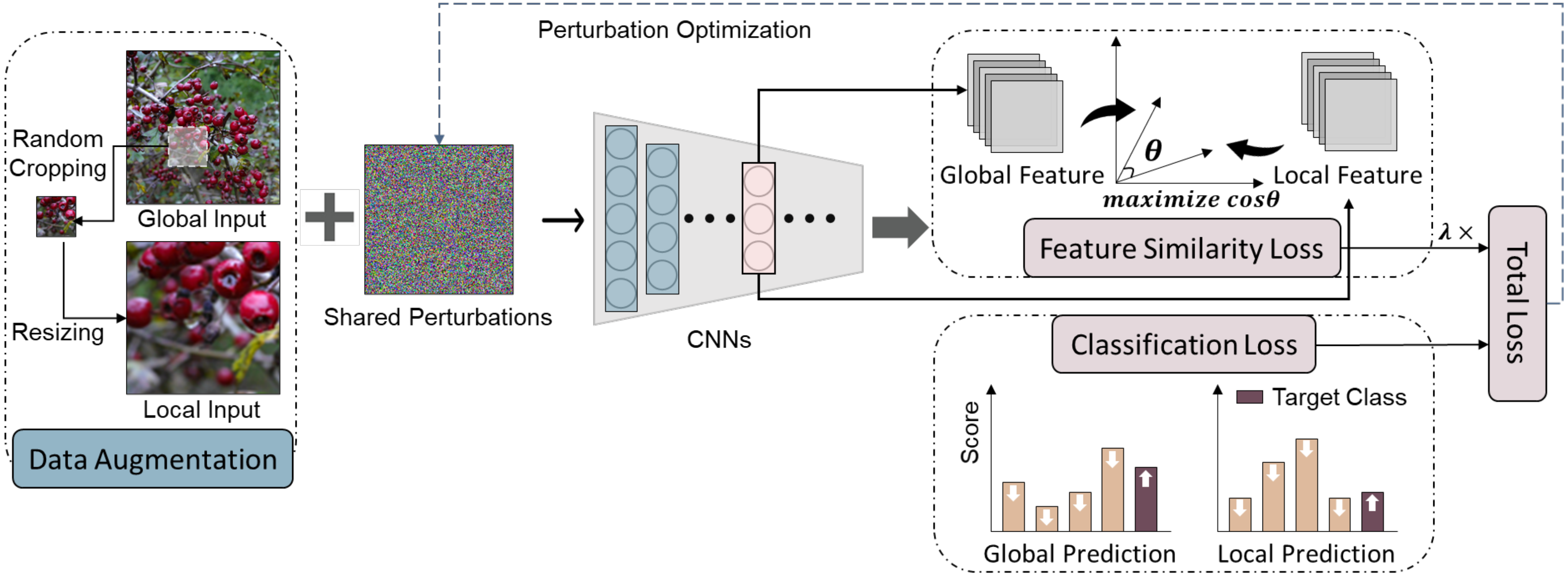
Zhipeng Wei, Jingjing Chen, Zuxuan Wu, Yu-Gang Jiang

Shanghai Key Lab of Intelligent Information Processing, School of Computer Science, Fudan University
Shanghai Collaborative Innovation Center of Intelligent Visual Computing
WED-AM-387

Quick Preview

The proposed Self-Universality Attack

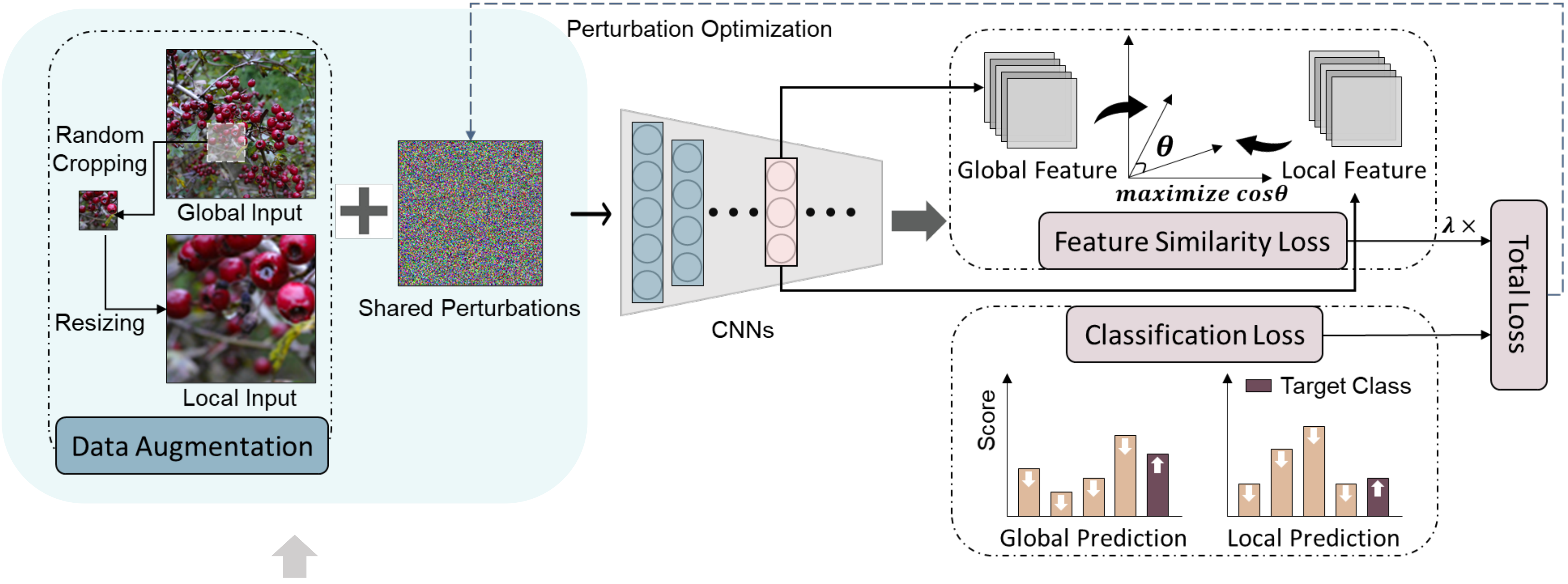
- Our Self-Universality method optimizes the perturbation to be agnostic to different local regions within one image, which is called self-universality.



Quick Preview

The proposed Self-Universality Attack

- Our Self-Universality method optimizes the perturbation to be agnostic to different local regions within one image, which is called self-universality.

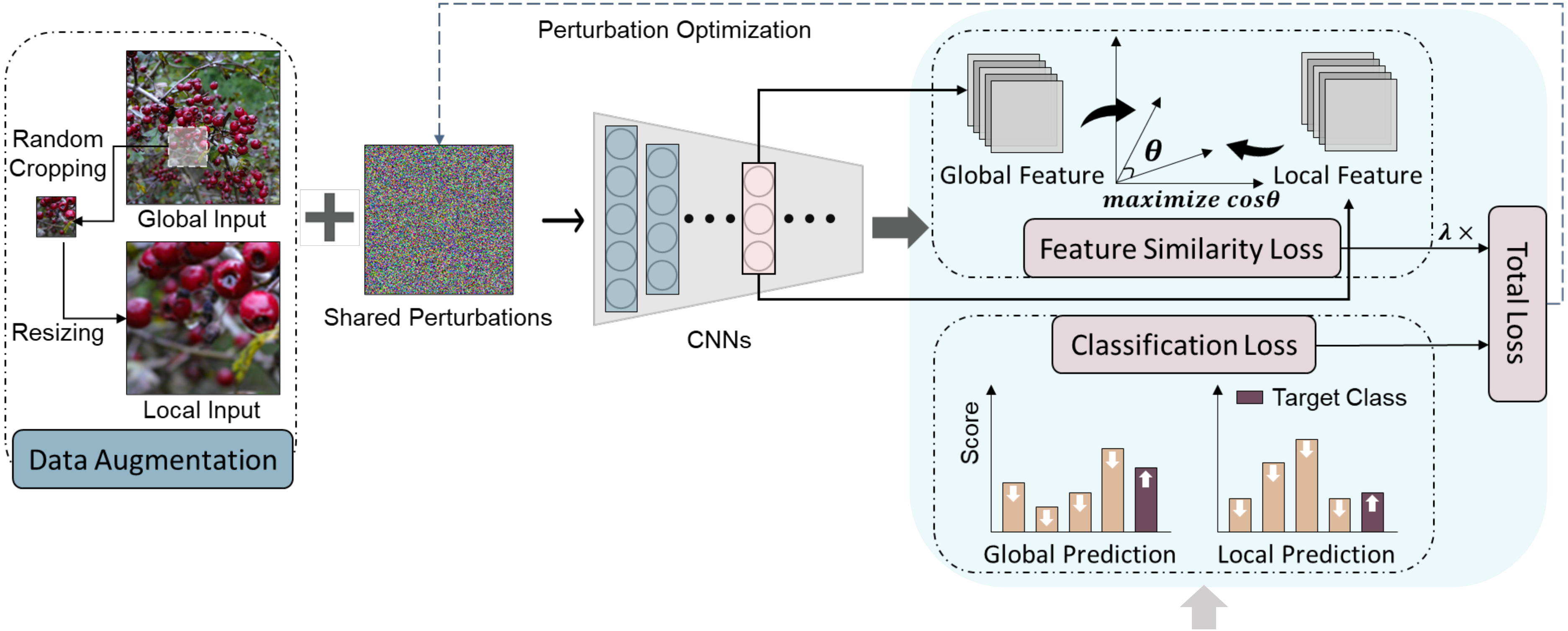


Incorporating randomly cropped local regions within one image into the iterative attacks.

Quick Preview

The proposed Self-Universality Attack

- Our Self-Universality method optimizes the perturbation to be agnostic to different local regions within one image, which is called self-universality.



Generate perturbations with more dominant features by maximize the cosine similarity of intermediate features between the adversarial global and local inputs.

Outline

Background and Motivation

- Transferable Targeted Attacks
- Universality of Targeted Perturbations

Methodology

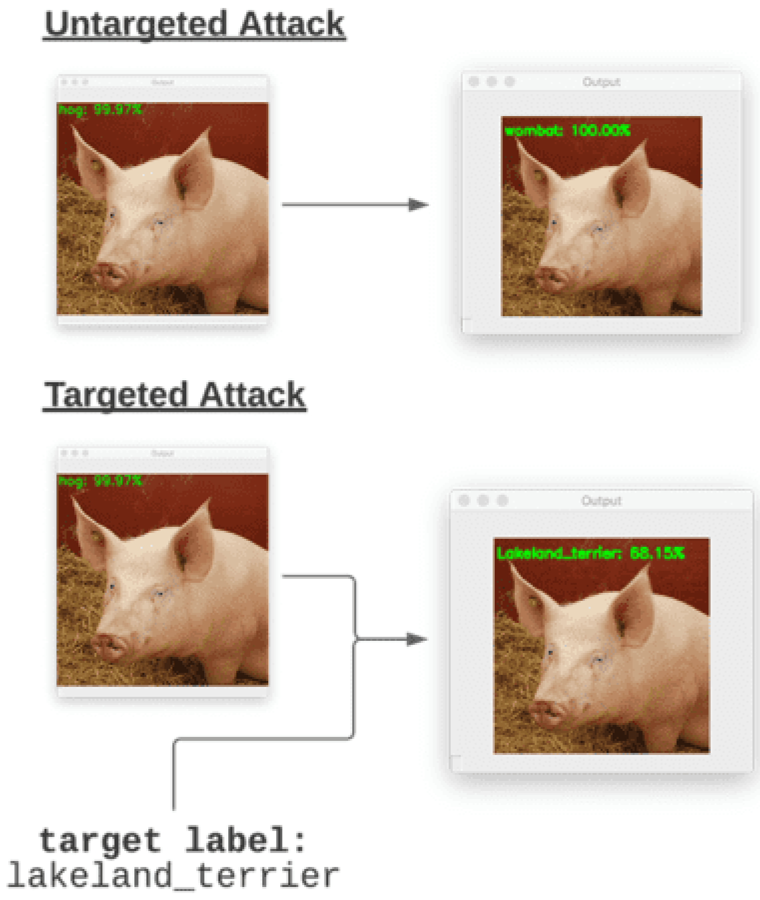
- Self-Universality (SU) Attack

Experiment

- Single-model transferable attacks
- Ensemble model transferable attacks
- Combination with existing methods
- Ablation Study

Background

Targeted Attacks vs. Untargeted Attacks



Untargeted Attacks: no control over the output class label

Targeted Attacks: incorporate label information into the optimization.

https://pyimagesearch.com/2020/10/26/targeted-adversarial-attacks-with-keras-and-tensorflow/

Background

Transferable Targeted Attacks

Resource-intensive methods

- Training target-class-specific classifiers (FDA[1][2])
- Training target-class-specific generators (TTP[3])

Iterative methods

- A large iteration and Logit loss (Logit[4])
- Rendering image on a 3D object (ODI[5])

[1] Nathan Inkawhich, Kevin Liang, Binghui Wang, Matthew Inkawhich, Lawrence Carin, and Yiran Chen. Perturbing across the feature hierarchy to improve standard and strict blackbox attack transferability. Advances in Neural Information Processing Systems, 33:20791–20801, 2020.

[2] Nathan Inkawhich, Kevin J Liang, Lawrence Carin, and Yiran Chen. Transferable perturbations of deep feature distributions. arXiv preprint arXiv:2004.12519, 2020.

[3] Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. On generating transferable targeted perturbations. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 7708–7717, 2021.

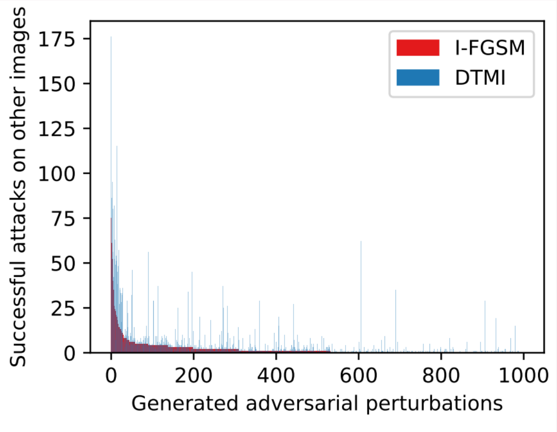
[4] Zhengyu Zhao, Zhuoran Liu, and Martha Larson. On success and simplicity: A second look at transferable targeted attacks. Advances in Neural Information Processing Systems, 34:6115–6128, 2021.

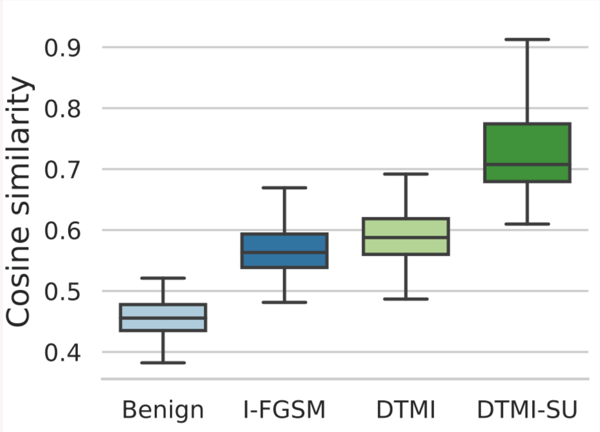
[5] Junyoung Byun, Seungju Cho, Myung-Joon Kwon, Hee-Seon Kim, and Changick Kim. Improving the transferability of targeted adversarial examples through object-based diverse input. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 15244–15253, 2022.

Refer to https://slideslive.com/38967912/on-success-and-simplicity-a-second-look-at-transferable-targeted-attacks?ref=recommended

Motivation

Universality of Targeted Perturbations





- There is a relatively positive correlation between universality and targeted transferability.
- Targeted perturbations produce more dominant features.
- Adversarial examples with high universality tend to be more transferable in targeted attacks.

https://pyimagesearch.com/2020/10/26/targeted-adversarial-attacks-with-keras-and-tensorflow/

Outline

Background and Motivation

- Transferable Targeted Attacks
- Universality of Targeted Perturbations

Methodology

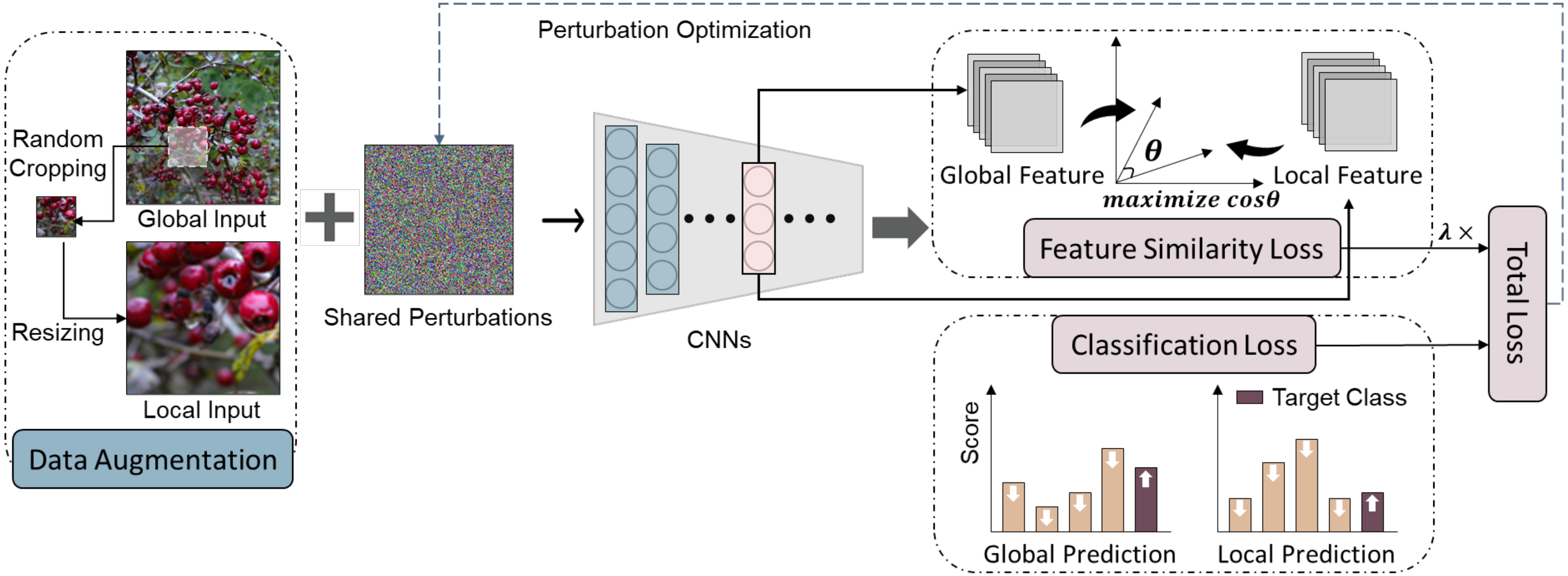
- Self-Universality (SU) Attack

Experiment

- Single-model transferable attacks
- Ensemble model transferable attacks
- Combination with existing methods
- Ablation Study

Methodology

The proposed Self-Universality Attack



Methodology

The proposed Self-Universality Attack

Algorithm 1 DTMI-SU attack

Input: the classification loss function J, white-box model f, benign image x, targeted class y_t.
Parameter: The perturbation budget ϵ, iteration number I, step size α, scale parameter $s = \{s_l, s_{int}\}$, weighted parameter λ, and DTMI parameters $T(\cdot, p)$, W, μ.
Output: The adversarial example x_{adv}.

Initialize δ_0 and g_0 by Eq.1
for $i = 0$ to $I - 1$ **do**
  Random cropping and resizing: $\hat{x} = Loc(x, s)$
  DI: $x' = T(x + \delta_i, p)$; $\hat{x}' = T(\hat{x} + \delta_i, p)$
  Calculate gradients: $g_{i+1} = \nabla_\delta(J(f(x'), y_t) + J(f(\hat{x}'), y_t) - \lambda \cdot CS(f_l(x'), f_l(\hat{x}')))$
  $g_{i+1} = \mu \cdot g_i + \frac{W \cdot g_{i+1}}{||W \cdot g_{i+1}||_1}$
  Update and Clip δ_{i+1} by Eq.3, 4
end for
return $x + \delta_I$

Outline

Background and Motivation

- Transferable Targeted Attacks
- Universality of Targeted Perturbations

Methodology

- Self-Universality (SU) Attack

Experiment

- Single-model transferable attacks
- Ensemble model transferable attacks
- Combination with existing methods
- Ablation Study

Experiment

Single-model transferable attacks

Attack	White-box Model: Res50			White-box Model: Dense121		
	→ Dense121	→ VGG16	→ Inc-v3	→ Res50	→ VGG16	→ Inc-v3
DTMI-CE	27.1/39.7/44.3	18.9/27.6/29.4	2.2/3.4/4.1	12.9/16.7/18.4	8.1/10.6/10.6	1.7/2.2/3.2
DTMI-CE-SU	6.2/27.8/**54.2**	3.0/20.2/**45.4**	0.2/4.5/**10.1**	2.6/17.6/**39.4**	1.4/12.6/**32.4**	0.2/4.8/**10.8**
DTMI-Logit	30.4/64.4/71.8	22.6/55.1/62.8	2.7/7.1/9.6	16.1/39.3/43.7	13.5/33.0/38.1	2.1/7.1/7.7
DTMI-Logit-SU	23.8/63.9/**75.5**	16.6/55.9/**66.9**	2.0/8.3/**11.6**	12.8/42.9/**50.2**	9.3/37.2/**45.2**	1.8/7.5/**10.4**
Attack	**White-box Model: VGG16**			**White-box Model: Inc-v3**		
	→ Res50	→ Dense121	→ Inc-v3	→ Res50	→ Dense121	→ VGG16
DTMI-CE	0.6/0.6/0.5	0.4/0.3/0.4	0.0/0.0/0.0	0.8/1.8/2.4	0.8/2.4/2.9	0.7/1.3/1.8
DTMI-CE-SU	0.2/2.1/**2.8**	0.2/2.1/**3.2**	0.0/0.2/**0.2**	0.4/1.2/**2.9**	0.2/1.4/**5.0**	0.1/0.8/**2.5**
DTMI-Logit	3.0/9.6/11.3	3.2/12.0/13.7	0.1/0.6/0.7	0.9/2.0/2.8	1.1/3.3/5.0	0.6/2.2/3.9
DTMI-Logit-SU	3.4/11.7/**13.9**	3.3/13.8/**16.1**	0.2/0.9/**0.8**	0.6/2.3/**4.3**	0.5/3.8/**7.4**	0.3/1.9/**4.4**

Table 2. TASR (%) of all black-box models under four attack scenarios using ResNet50, DenseNet121, VGGNet16 and Inception-v3 as white-box models, respectively. We conduct these experiments three times and report average TASR with 20/100/300 iterations, the standard deviation is shown in Appendix. The best results with 300 iterations are in bold.

Experiment

Ensemble model transferable attacks

Ensemble Attack	Black-box Model				Average
	Res50	Dense121	VGG16	Inc-v3	
DTMI-CE	31.1	55.2	51.6	16.1	38.5
DTMI-CE-SU	**55.7**	**65.0**	**68.2**	**29.3**	**54.5**
DTMI-Logit	70.2	82.3	82.2	29.1	65.9
DTMI-Logit-SU	**75.3**	**82.9**	**84.2**	**34.5**	**69.2**

Table 3. TASR (%) of one black-box model in ensemble transfer attacks. TASR with 300 iterations is reported. The best results are in bold.

Experiment

Combination with existing methods

Attack	Dense121	VGG16	Inc-v3
DTMI-SI/+SU	85.7/**87.2**	69.0/**71.8**	35.8/**41.6**
DTMI-Adm./+SU	89.1/**89.4**	75.7/**79.1**	42.1/**47.1**
DTMI-EMI/+SU	71.0/**79.0**	64.6/**82.4**	5.0/**14.8**
ODI-TMI/+SU	89.9/**92.8**	81.0/**91.7**	66.9/**72.0**

Table 4. Average TASR (%) of different combinational attacks. We use ResNet50 as the white-box model, and report the results with 300 iterations. Logit is used here.

Ablation Study

Local	Feature Similarity Loss	Averaged TASR
-	-	9.8
✓	-	10.9
✓	✓	15.6

Table 5. Average TASR (%) of black-box models for our proposed method with different component combinations. The classification loss is set as CE. '✓' indicates that the component is used while '-' indicates that it is not used. TASR is averaged among four attack scenarios using ResNet50, DenseNet121, VGGNet16 and Inceptionv3 as white-box models, respectively.

Thank you!

Code

Paper