PatchAttack: A Black-box Texture-based Attack with Reinforcement Learning

Chenglin Yang, Adam Kortylewski, Cihang Xie, Yinzhi Cao, Alan Yuille

CCVL, Johns Hopkins University

Motivation

It is known that digital perturbations can easily fool the deep network.

(FGSM, PGD, C&W, ...)

A. Kurakin et. al., Adversarial Examples in the Physical World, ICLR 2017

Motivation

It is known that digital perturbations can easily fool the deep network.

(FGSM, PGD, C&W, ...)

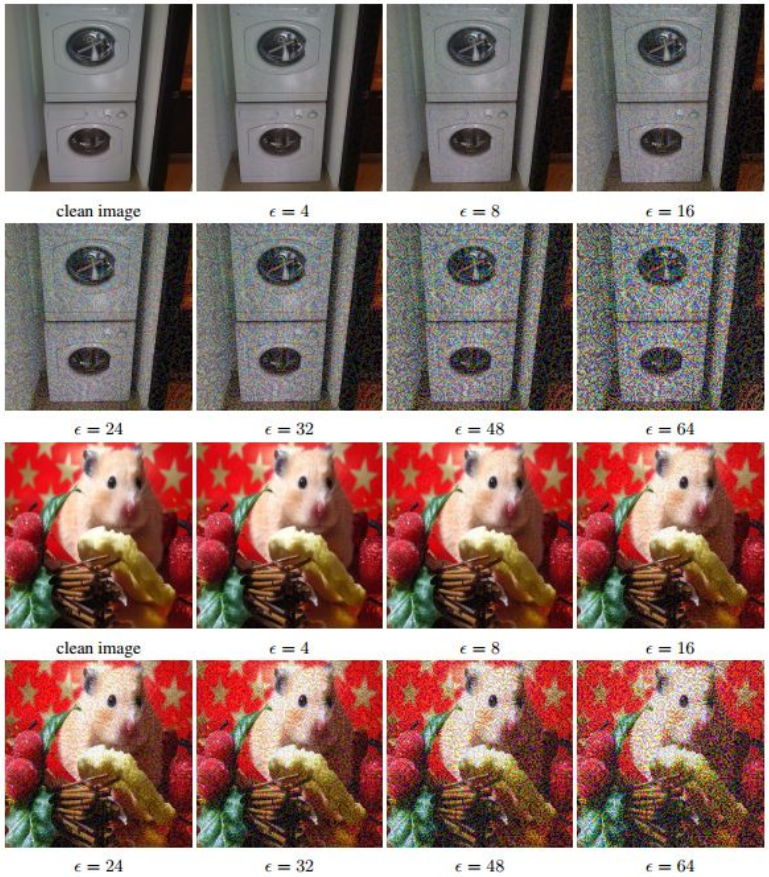
These type of attacks are well investigated, and **not very interesting** these days.

A. Kurakin et. al., Adversarial Examples in the Physical World, ICLR 2017

Motivation

Easy tasks for adversaries

1. **Know** both the architectures and weights.
2. Backpropagate the gradients to all the pixels according to intuitive loss functions.



A. Kurakin et. al., Adversarial Examples in the Physical World, ICLR 2017

Motivation

Let’s jump out and have a overview of the different attacks:

A. **Gradient-based (white-box) attack:**
   a. **Global perturbations** (**discussed in previous slides**)
   b. Local perturbations
      i. Adversarial Patch
      ii. UPC

B. Gradient-free (Black-box) attack:
   a. Global perturbations
      i. ZOO, NES, Bandits, GenAttack...
   b. Local perturbations
      i. Ours

Motivation

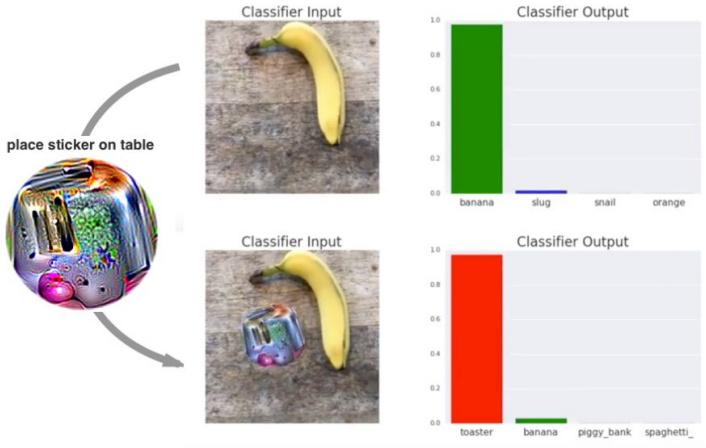
Let’s jump out and have a overview of the different attacks:

A. Gradient-based (white-box) attack:

- a. Global perturbations (discussed in previous slides)
- b. Local perturbations
  - **i. Adversarial Patch**
  - ii. UPC

B. Gradient-free (Black-box) attack:

- a. Global perturbations
  - i. ZOO, NES, Bandits, GenAttack...
- b. Local perturbations
  - i. Ours



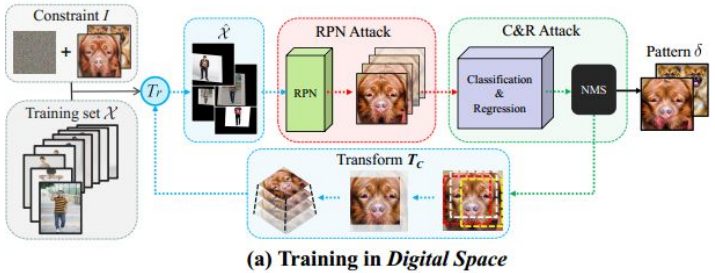
T.B. Brown et. al., Adversarial Patch, arXiv preprint 2017

Motivation

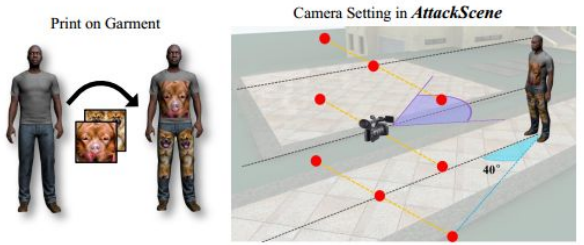
Let's jump out and have a overview of the different attacks:

A. Gradient-based (white-box) attack:
- a. Global perturbations (discussed in previous slides)
- b. Local perturbations
  - i. Adversarial Patch
  - **ii. UPC**

B. Gradient-free (Black-box) attack:
- a. Global perturbations
  - i. ZOO, NES, Bandits, GenAttack...
- b. Local perturbations
  - i. Ours



(a) Training in *Digital Space*



(b) Attacking in *Physical Space*

L. Huang et. al., Universal Physical Camouflage Attacks on Object Detectors, CVPR 2020

Motivation

Let’s jump out and have a overview of the different attacks:

A. Gradient-based (white-box) attack:
   a. Global perturbations (discussed in previous slides)
   b. Local perturbations
      i. Adversarial Patch
      ii. UPC

B. Gradient-free (Black-box) attack:
   a. Global perturbations
      i. ZOO, NES, Bandits, GenAttack… Finite Difference Gradient estimation, Evolution Alg.
   b. Local perturbations
      i. Ours

Motivation

Let’s jump out and have a overview of the different attacks:

A. Gradient-based (white-box) attack:
   a. Global perturbations (discussed in previous slides)
   b. Local perturbations
      i. Adversarial Patch
      ii. UPC

B. Gradient-free (Black-box) attack:
   a. Global perturbations
      i. ZOO, NES, Bandits, GenAttack...
   b. Local perturbations
      i. Ours PatchAttack

Defining Patch-based Attack

Mathematical Framework:

$$\mathcal{L}\left(\mathbf{y}, y'\right), \quad \text{where} \quad \mathbf{y} = \mathbf{f}\left(\mathbf{g}\left(\mathbf{x}\right); \boldsymbol{\theta}\right), \tag{1}$$

$$\mathbf{g}\left(\mathbf{x}\right) : \begin{cases} x_{u,v} = \mathbf{T}\left(x_{u,v}\right), & \text{if} \quad (u,v) \in \mathcal{E} \\ x_{u,v} = x_{u,v}, & \text{otherwise} \end{cases} \tag{2}$$

$$\mathcal{E} = \mathbf{s}\left(\mathbf{x}, \mathbf{f}(\cdot, \theta), \mathcal{S}\right) \subseteq \left\{(u,v) \,|u \in [0,H), v \in [0,W)\right\} \tag{3}$$

Defining Patch-based Attack

Intuitive Explanation:

A. Optimize a Image-specific location to superimpose the patch
B. Optimize the Image-specific pattern of this patch
C. Simultaneously and in a non-differential process

Sampled-based Attack

Metropolis-Hasting sampling:

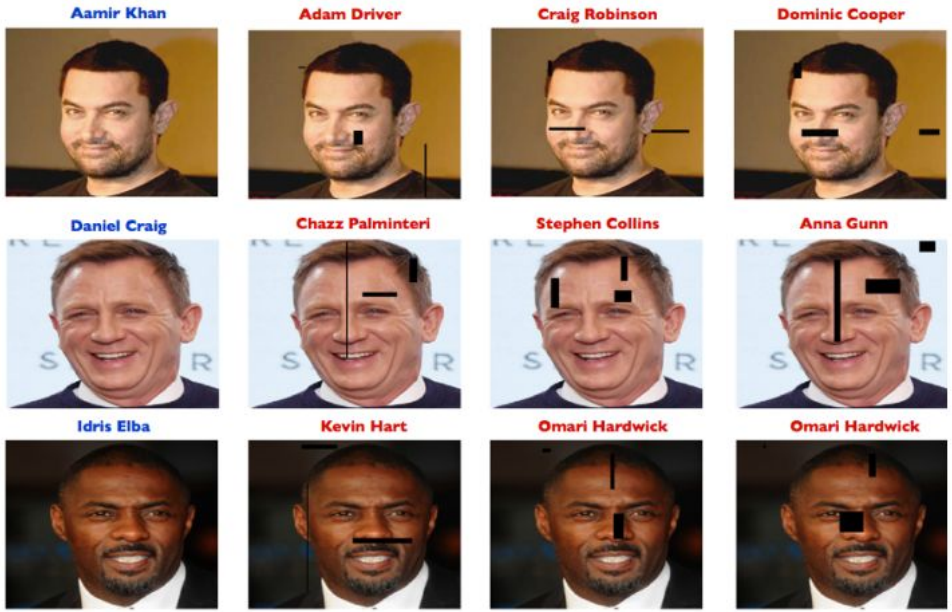
Effective non-target attack in the fine-grained task, e.g., face recognition

- Not powerful

  Not effective in targeted-attack scenario: either the occlusion area is too large or the failure attack occurs

- Not efficient:

  Large-number of queries are required



A. Fawzi et. al., Measuring the effect of nuisance variables on classifiers, BMVC 2016

PatchAttack

In our PatchAttack, we model the attack as a decision-making process where an agent finds the best position in the image to superimpose the patches and the way how to texture them through reinforcement learning.

PatchAttack

Monochrome Patch Attack (MPA):

MPA_Gray: Optimize the patch locations and zero out the pixel values of the patch

MPA_RGB: Optimize the patch locations and colorize the patches

Texture-based Patch Attack (TPA):

TPA: Optimize the patch locations and texture the patches

PatchAttack: MPA

Patch Search with Reinforcement Learning:

$$\mathcal{S} = \{(u_1^1, v_1^2, u_1^3, v_1^4, \cdots, u_C^1, v_C^2, u_C^3, v_C^4)\} \tag{4}$$

$$\mathbb{A}(\theta_{\mathbb{A}}) : P(a_t|(a_1, \cdots, a_{t-1}), \mathbf{f}(\cdot;\theta), \mathbf{x}) \qquad t = \{1, \cdots, 4C\} \tag{5}$$

$$\mathbf{r} = \begin{cases} \ln y' - \mathbf{A}(\mathbf{a})/\sigma^2, & \text{target attack} \\ \ln(1-y) - \mathbf{A}(\mathbf{a})/\sigma^2, & \text{non-target attack} \end{cases} \tag{6}$$

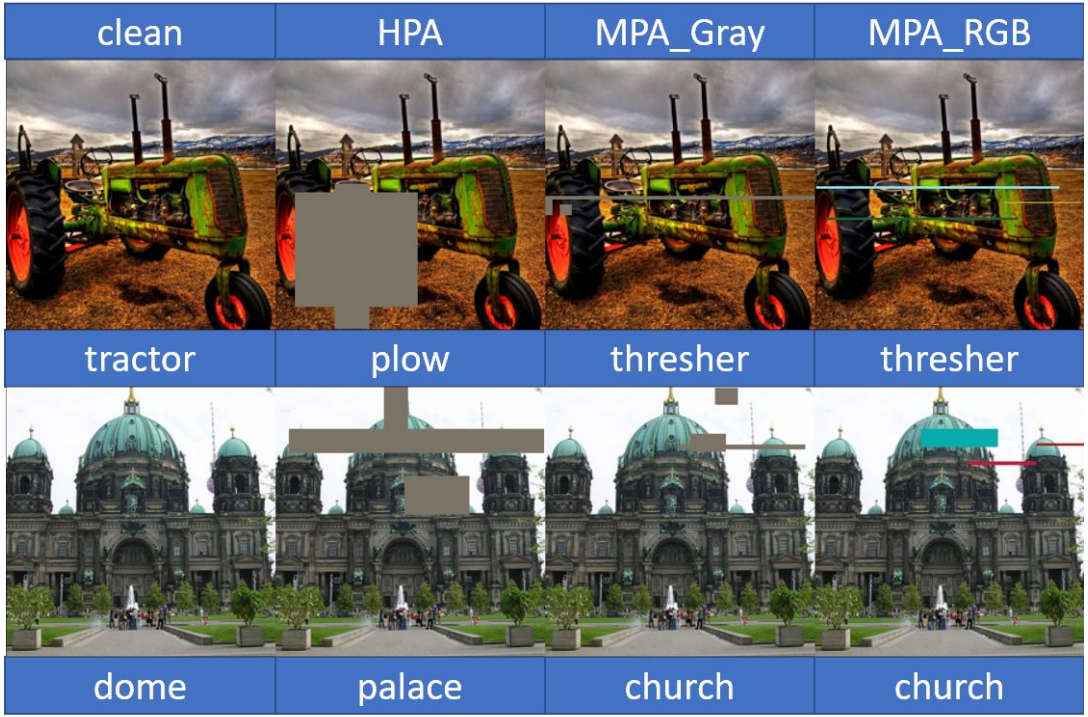
$$\text{MPA} : \begin{cases} \mathcal{E} = \mathbf{J}(\mathbf{a}) \\ \mathbf{T}(x_{u,v}) = 0 \\ \mathcal{L} = -\mathbf{r} \cdot \ln \mathbf{P} \end{cases} \tag{7}$$

$$\mathcal{S} = \{(u_1^1, v_1^2, u_1^3, v_1^4, R_1^5, G_1^6, B_1^7, \cdots, u_C^1, v_C^2, u_C^3, v_C^4, R_C^5, G_C^6, B_C^7)\} \tag{8}$$

PatchAttack: MPA

MPAs are powerful in non-targeted setting.

PatchAttack: MPA

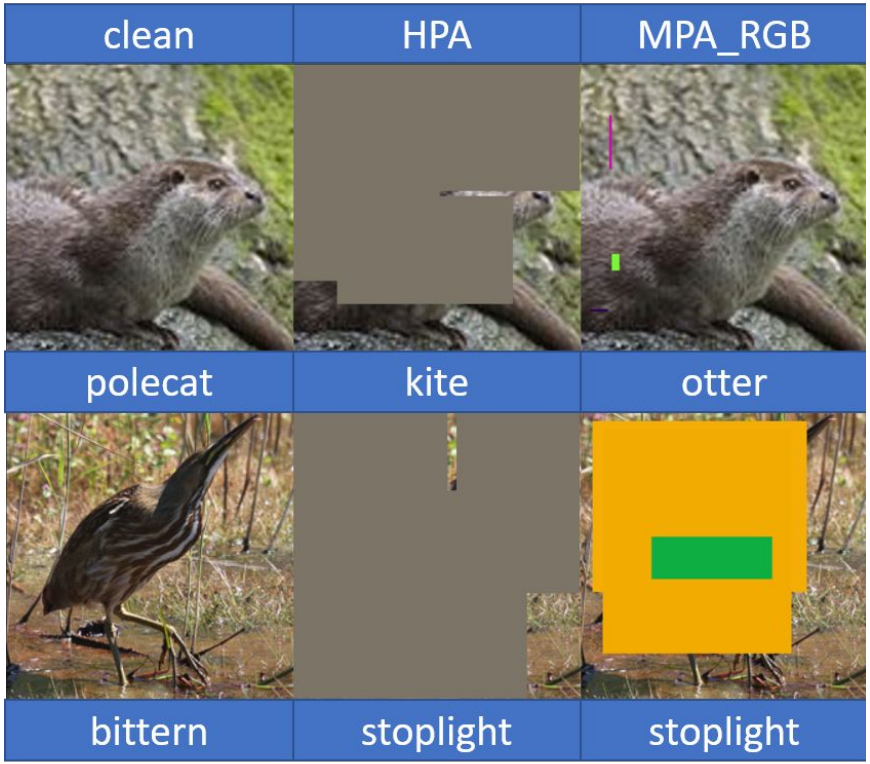


PatchAttack: MPA

MPAs are powerful in non-targeted setting.

But not satisfying in targeted-setting.

PatchAttack: MPA



PatchAttack: from MPA to TPA

MPAs are powerful in non-targeted setting, but not satisfying in targeted-setting.

Reason: MPAs only switch off the information on some parts of the image instead of adding additional information, which prevents it from performing targeted attacks. For example, MPA_RGB achieves superior performance compared with MPA_Gray.

PatchAttack: from MPA to TPA

MPAs are powerful in non-targeted setting, but not satisfying in targeted-setting.

Reason: MPAs only switch off the information on some parts of the image instead of adding additional information, which prevents it from performing targeted attacks. For example, MPA_RGB achieves superior performance compared with MPA_Gray.

Solution: Enable the reinforcement learner not only to find where to put the patch but also to figure out how to texture the patch. The core problem is to find an efficient parameterization of the texture, in order to retain fast and query efficient attacks.

PatchAttack: from MPA to TPA

MPAs are powerful in non-targeted setting, but not satisfying in targeted-setting.

Reason: MPAs only switch off the information on some parts of the image instead of adding additional information, which prevents it from performing targeted attacks. For example, MPA_RGB achieves superior performance compared with MPA_Gray.

Solution: Enable the reinforcement learner not only to find where to put the patch but also to figure out how to texture the patch. The core problem is to find an efficient parameterization of the texture, in order to retain fast and query efficient attacks.

We build a class-specific texture dictionary.

PatchAttack: Texture Dictionary

Style Transfer:

Content:

Feature maps tensors F^l

Style of an Image

$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk}$$

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} \left(G^l_{ij} - \hat{G}^l_{ij} \right)^2$$

$$\mathcal{L}(\vec{x}, \hat{\vec{x}}) = \sum_{l=0}^{L} w_l E_l$$

L.A. Gatys et. al., Image Style Transfer Using Convolutional Neural Networks, CVPR 2016

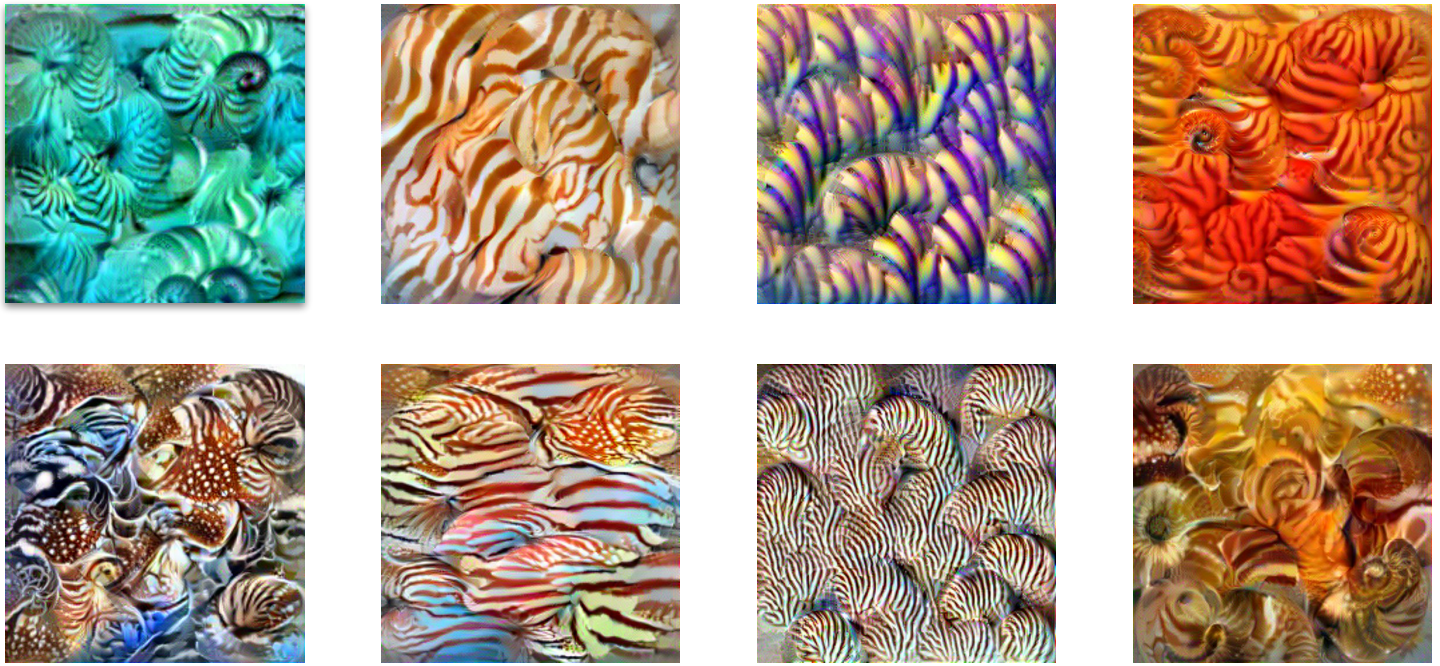
PatchAttack: Texture Dictionary

Procedures of generating texture images in the dictionary

- ➢ Collect Images of one specified class
- ➢ Use Grad-CAM to filter the important spatial locations
- ➢ Extract Styles
- ➢ Use k-means clustering to calculate 30 texture embeddings
- ➢ Generate texture images from texture embeddings

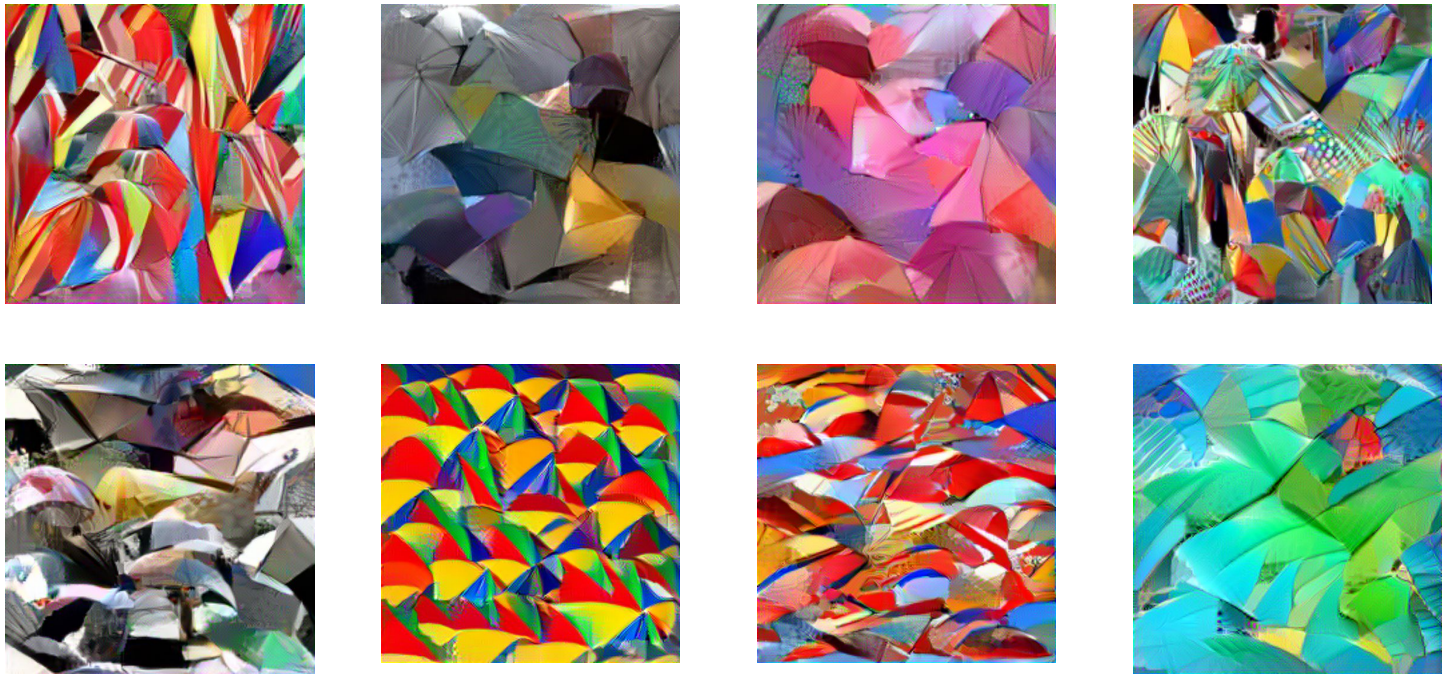
1,000 classes, 30,000 texture images, build upon the training set of ImageNet

PatchAttack: Texture Dictionary



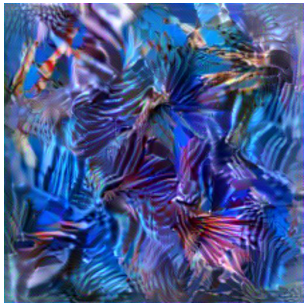
Nautilus

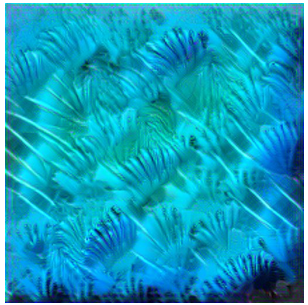
PatchAttack: Texture Dictionary

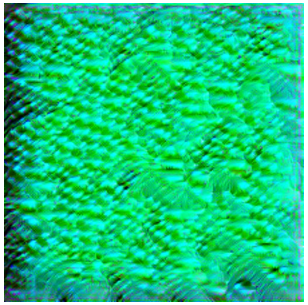


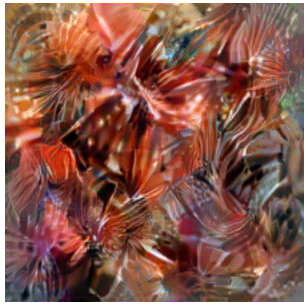
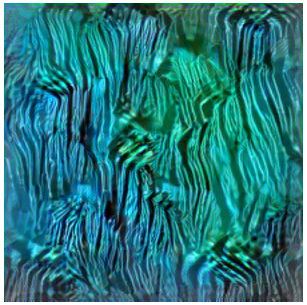
Umbrella

PatchAttack: Texture Dictionary







Lionfish

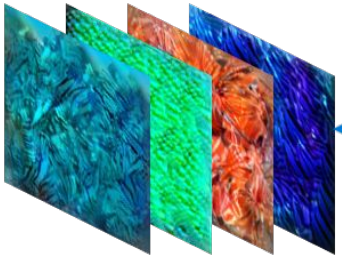
PatchAttack: TPA

Integrating the Texture Dictionary into Patch Attack

$$\mathcal{S} = \{(u_1^1, v_1^2, i_1^3, u_1^4, v_1^5, \cdots, u_C^1, v_C^2, i_C^3, u_C^4, v_C^5)\} \tag{9}$$

$$\text{TPA}: \begin{cases} \mathcal{E} = \mathbf{J}_t^1(u_1^1, v_1^2, \cdots, u_C^1, v_C^2) \\ \mathbf{T}\left(x_{u,v}\right) = \mathbf{J}_t^2((i_1^3, u_1^4, v_1^5, \cdots, i_C^3, u_C^4, v_C^5)) \end{cases} \tag{10}$$

PatchAttack: TPA



(2) Select texture image

Texture dictionary for the target class

RL Agent

(6) Get reward

DCNN

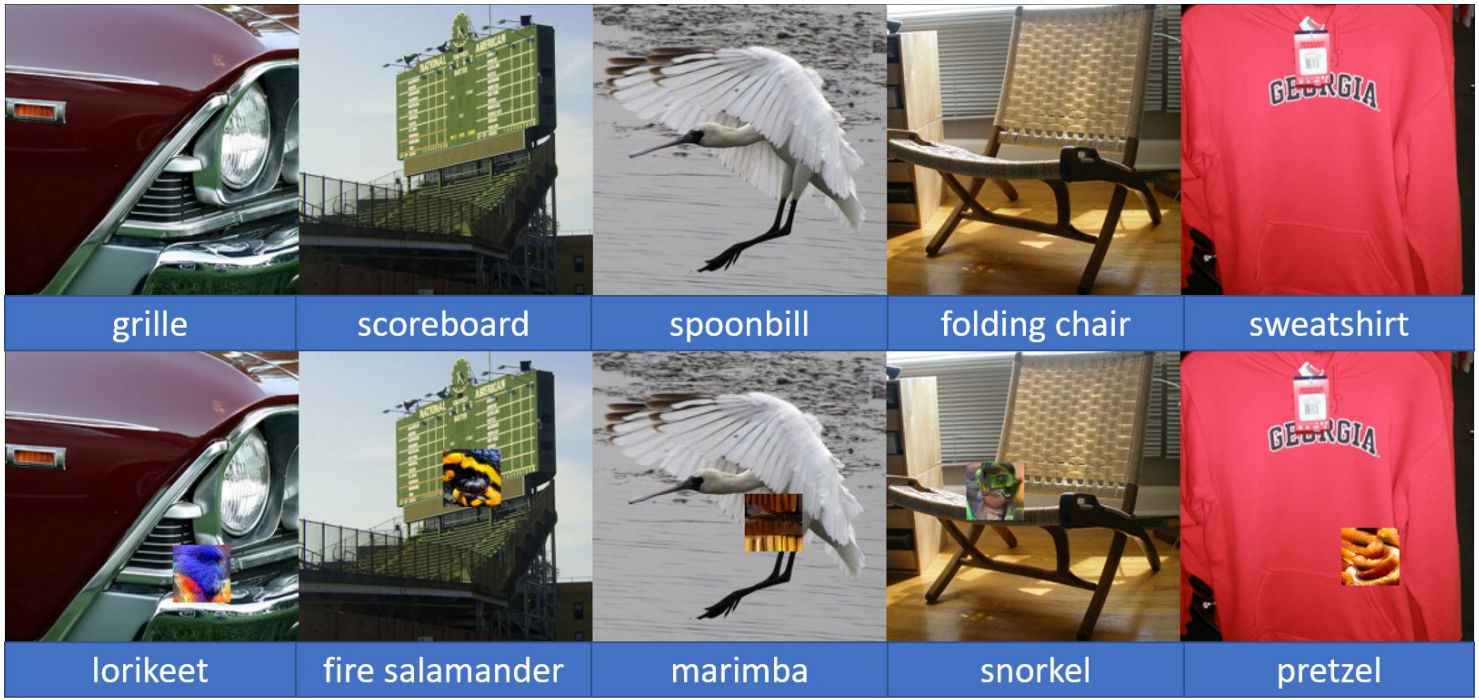
(1) Select the patch position in the input

(3) Select the patch position in the texture image

(4) Texture the patch

(5) Feed the image to DCNN

PatchAttack: TPA



Experiments

Non-targeted Attack

1000 images randomly selected from the validation set of ImageNet

Network	Attack	Acc. (%)	Avg_area (%)	Avg_qry
ResNet50	–	72.80	–	–
	HPA	0.40	18.05	10000
	MPA_Gray	0.00	6.57	9659
	MPA_RGB	0.00	5.41	9681
	TPA_N4_4%	0.30	5.06	1137
	TPA_N8_2%	0.30	3.10	983
DenseNet121	–	74.10	–	–
	HPA	0.10	19.82	10000
	MPA_Gray	0.00	6.87	9624
	MPA_RGB	0.00	5.73	9696
	TPA_N4_4%	0.50	5.13	1195
	TPA_N8_2%	0.30	3.13	1001
ResNeXt50	–	76.20	–	–
	HPA	0.80	19.22	10000
	MPA_Gray	0.00	7.88	9748
	MPA_RGB	0.00	6.23	9752
	TPA_N4_4%	0.70	5.21	1280
	TPA_N8_2%	0.50	3.25	1088
MobileNet-V2	–	68.80	–	–
	HPA	0.20	16.61	10000
	MPA_Gray	0.00	5.35	9578
	MPA_RGB	0.00	4.11	9603
	TPA_N4_4%	0.30	4.63	862
	TPA_N8_2%	0.30	2.74	756

Experiments

Targeted Attack

1000 images randomly selected from the validation set of ImageNet

Target labels are randomly selected

Network	Attack	T_acc. (%)	Avg_area (%)	Avg_qry
ResNet50	–	0.10	–	–
	HPA	23.20	71.54	50000
	MPA_RGB	25.90	18.45	28361
	TPA_N10_2%	97.60	7.80	15728
	TPA_N10_4%	99.70	9.97	8643
	TPA_N10_10%	100.00	15.36	3747
DenseNet121	–	0.10	–	–
	HPA	21.50	71.68	50000
	MPA_RGB	24.90	19.38	28088
	TPA_N10_2%	97.10	7.87	15920
	TPA_N10_4%	99.90	10.19	8953
	TPA_N10_10%	100.00	15.84	3970
ResNeXt50	–	0.00	–	–
	HPA	25.40	72.57	50000
	MPA_RGB	27.60	13.86	24738
	TPA_N10_2%	97.60	7.59	15189
	TPA_N10_4%	99.70	9.60	8223
	TPA_N10_10%	100.00	15.04	3538
MobileNet-V2	–	0.10	–	–
	HPA	22.10	69.45	50000
	MPA_RGB	27.70	16.64	28294
	TPA_N10_2%	98.50	7.78	15479
	TPA_N10_4%	99.90	10.39	8948
	TPA_N10_10%	100.00	16.85	4422

Experiments

Defense 1: Feature Denoising

Non-target	Attack	Acc. (%)	Avg_area (%)	Avg_qry
Denoise_ResNet152	–	61.60	–	–
	MPA_RGB	0.00	0.48	9287
	TPA_N4_4%	1.60	4.71	919
	TPA_N8_10%	1.30	2.91	867
Target	**Attack**	**T_acc. (%)**	**Avg_area (%)**	**Avg_qry**
Denoise_ResNet152	–	0.10	–	–
	MPA_RGB	38.30	6.39	27464
	TPA_N10_2%	84.00	9.73	22196
	TPA_N10_4%	94.60	13.40	13932
	TPA_N10_10%	99.30	20.90	6920

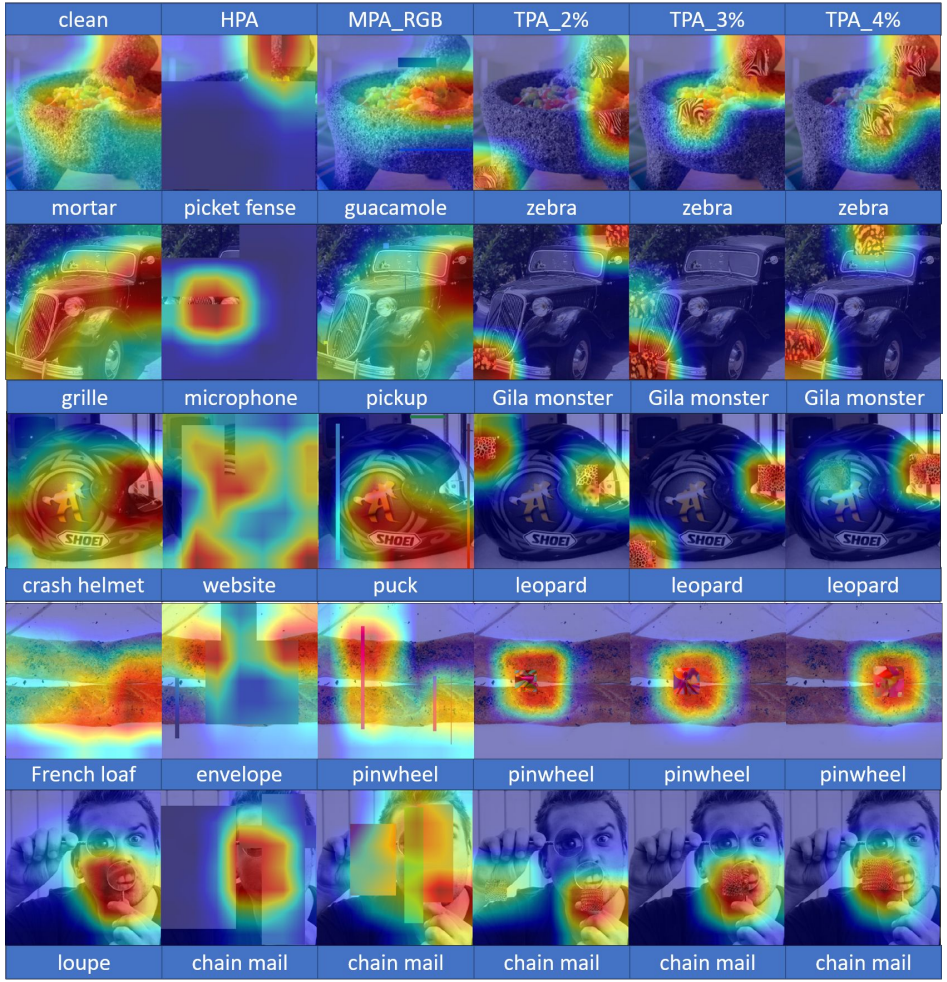
Experiments

Defense 2: Shape-biased Network

Non-target	Attack	Acc. (%)	Avg_area (%)	Avg_qry
Shape-Network	–	73.70	–	–
	TPA_N4_4%	0.50	5.19	1242
	TPA_N8_10%	0.20	3.17	1031
Target	**Attack**	**T_acc. (%)**	**Avg_area (%)**	**Avg_qry**
Shape-Network	–	0.10	–	–
	TPA_N10_2%	96.30	8.36	17443
	TPA_N10_4%	100.00	10.31	9229
	TPA_N10_10%	100.00	15.52	3822

Adversarial Examples

Attention Maps



Conclusion

We propose PatchAttack, a powerful black-box texture-based patch attack.

- Show that even small textured patches are able to break deep networks
- Monochrome Patch Attack achieves a strong performance on non-targeted attack, surpassing previous work by a large margin using less queries and smaller patch areas
- Texture-based Patch Attack achieves exceptional performance in both targeted and non-targeted attacks
- PatchAttack breaks traditional SOTA defenses and shape-based networks