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

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

CCVL, Johns Hopkins University

Patch-Attack: Summary

- Digital Perturbation Attacks which are imperceptible can fool Deep Nets. But many of these attacks are *white box,* i.e. require knowing the weights of the Deep Net, and there are a growing number of effective defenses.
- We develop a black box texture-based attack which requires no knowledge of the Deep Net. Our method learns an attack policy which selects texture-patches from a pre-defined dictionary.
- We show that we can successfully attack a range of Deep Nets for classification and reducing their effectiveness by up to 90% using only small patches.
- The attack is resistant to existing defenses.
- Chenglin Wang, Adam Kortylewski, Cihang Xie, Yinzhi Cao, Alan Yuille. “PatchAttack: A Black-box Texture-based Attack with Reinforcement Learning”. ECCV. In review. 2020.

Texture-Patch Attacks

Use Reinforcement Learning to learn a policy which, for each image, selects texture-patches and places them in the image so as to:

A. Optimize the position and scale of the patch(es)
B. Optimize the texture-pattern of the patch(es)

Mathematical Formulation:

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

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

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

PatchAttack: MPA

Patch Search with Reinforcement Learning:

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

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

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

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

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

Two types of Textures

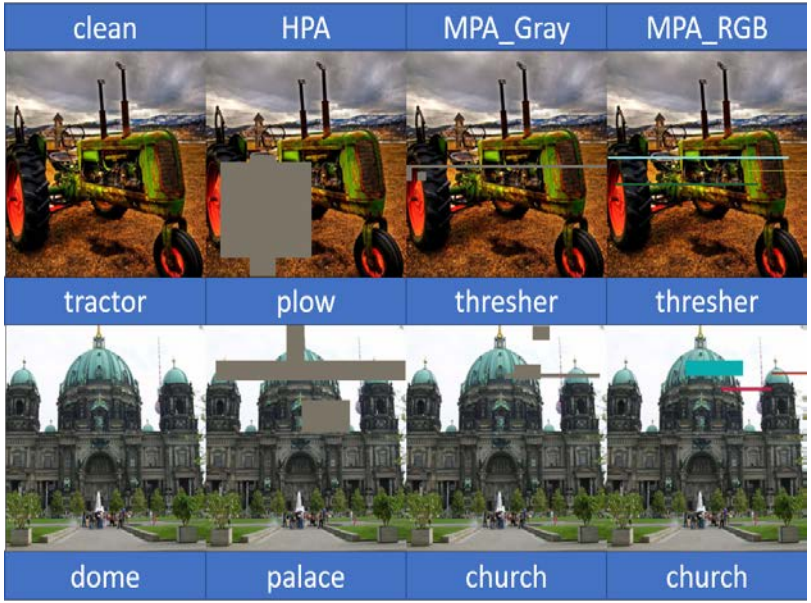
$$G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l$$

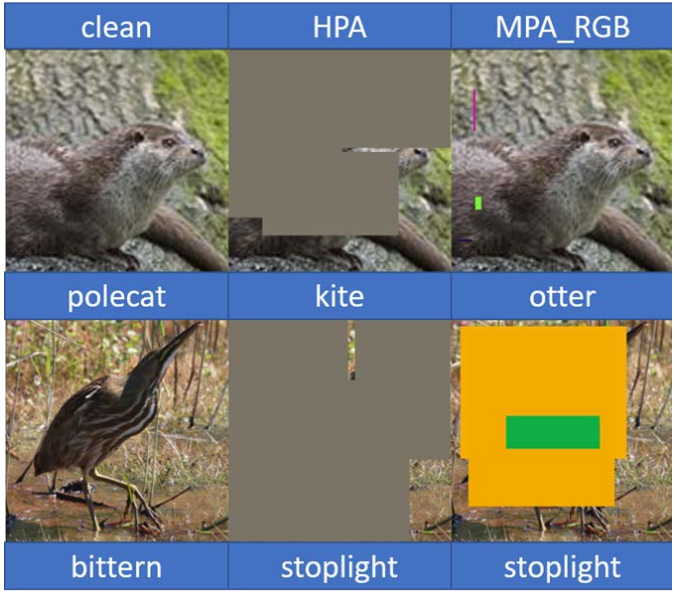
$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} \left(G_{ij}^l - \hat{G}_{ij}^l \right)^2$$

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

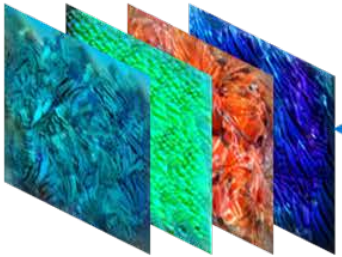
- (1). Monochrome Patch Attack (MPA):
- (2). Texture-Based Patch Attack (TPA):
- Texture Patches are learnt from a surrogate Deep Net (using GramMatrices).
- Nautilus. Umbrella. Lionfish.

MPA: small patches confuse similar object classes.





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

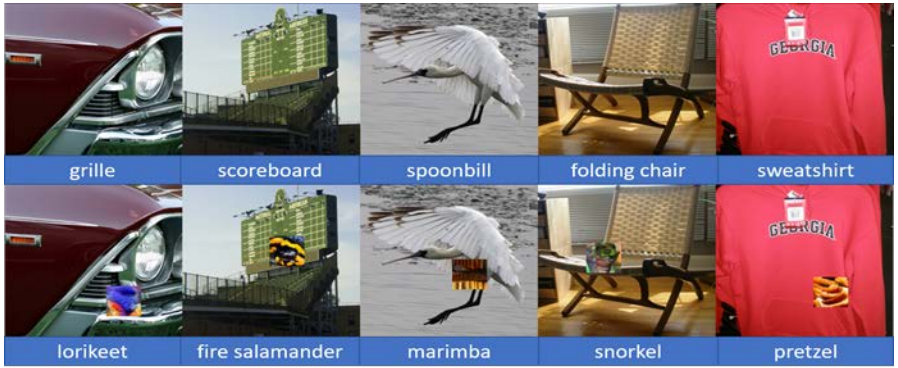
(3) Select the patch position in the texture image

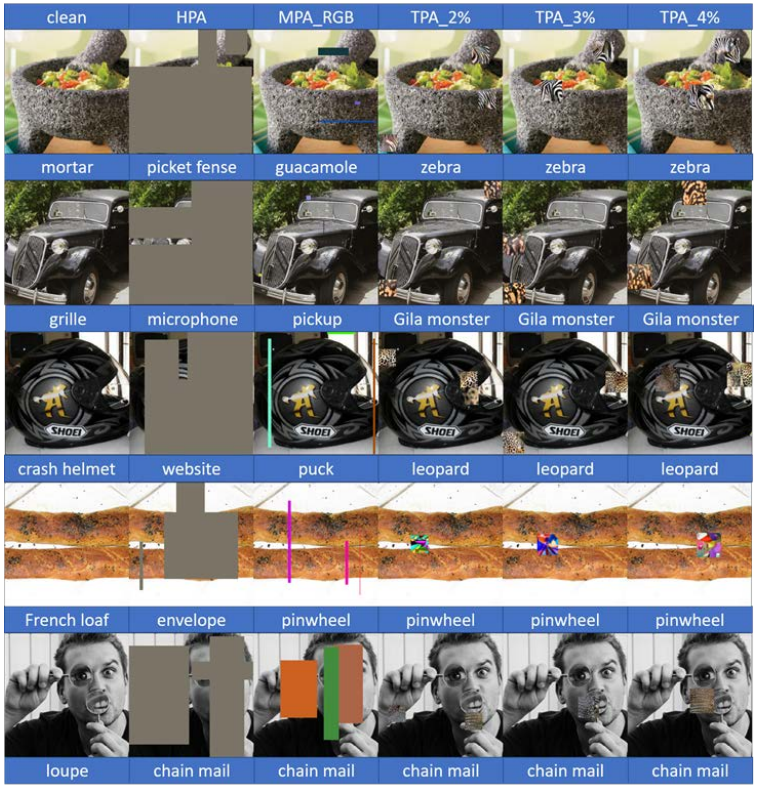
(4) Texture the patch

(5) Feed the image to DCNN

PatchAttack: TPA can confuse very different classes

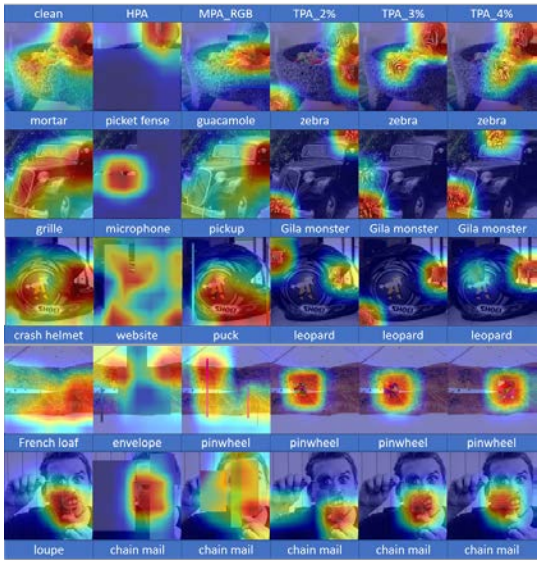
- Here are examples.





Intuition: Why Do Patch-Attacks Work?

- The Deep Net has an internal attention mechanism (buried in the weights). The Patch-Attack highjacks the attention mechanism so that it focuses on the patch. The Deep Net thinks the patch looks more like object itself.
- This shows a major limitation of Deep Networks.
- Recall that the texture-patches look like this:



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

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