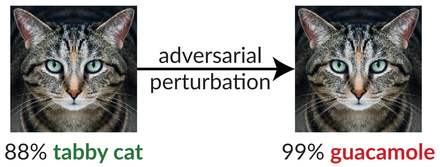
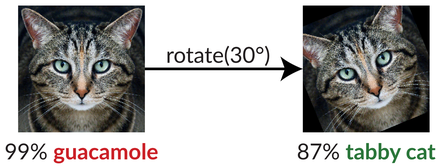
Synthesizing Robust Adversarial Examples

Anish Athalye*, Logan Engstrom*, Andrew Ilyas*, Kevin Kwok
Massachussets Institute of Technology, LabSix

Adversarial Examples are not Robust

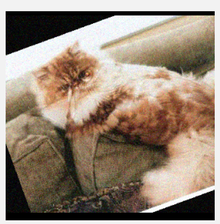
Adversarial examples that are synthesized using standard methods (e.g. FGSM, PGD) are not inherently robust to foveation and other image transformations.



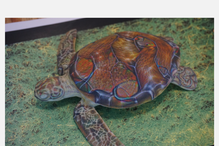


Robust Adversarial Examples

We can construct adversarial examples that are robust to a chosen distribution of transformations. This means:



2D Adversarial Examples that can be printed out and fool classifiers at any angle.



3D Adversarial Objects: For example, a model turtle with texture perturbed to be classified as rifle over most viewpoints.

Expectation Over Transformation

We introduce **Expectation Over Transformation (EOT)**, method for producing adversarial examples that are robust to transformation.

Key insight: optimize over a distribution of transformations instead of unmodified image

(x_0, y_t)	initial image, target class
$P(y \mid x)$	model output
T	transformation distribution
$d(\cdot, \cdot)$	distance metric
ϵ	perturbation bound

$\implies$ robust example x'

EOT finds a robust adversarial example x' by optimizing expected adversariality while bounding expected perturbation:

$$\arg\max_x \quad \mathbb{E}_{t\sim T}\left[P\left(y \mid t(x)\right)\right]$$

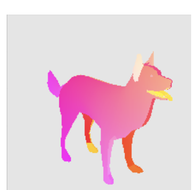
$$\text{subject to } \mathbb{E}_{t\sim T}\left[d(t(x), t(x_0))\right] < \epsilon$$

- To optimize, we take Lagrangian relaxation and use SGD
- Requires differentiation through classifier and transformation (chain rule)
- All transformations $t \in T$ must be differentiable

Constructing 3D Examples

- Can we optimize over random 3D transformations? (e.g. rotation, translation, scale, etc.)
- Recall that all transformations T must be differentiable
- We are only modifying texture of the model, not the model shape
- We can model $T(x)$ as a sparse matrix multiplication (map from coordinates in texture-space to render-space)

$\nabla_x T(x)$ is simply the generated render map



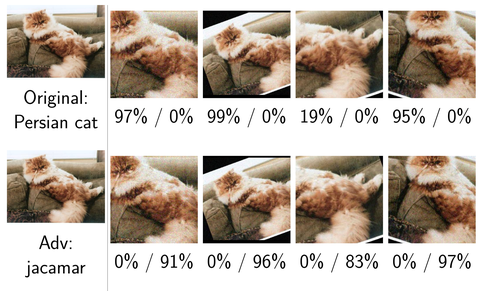
How we did it:

- Construct "RGB map" texture (x: R, y: G)
- Render texture on given model in some pose
- For each pixel in rendering, the color (R, G, B) encodes its position in the texture

2D Adversarial Attacks

We generate 1,000 2D adversarial examples and evaluate in simulation.

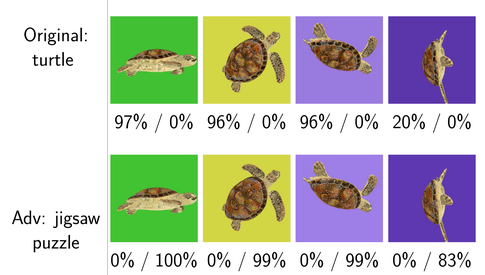
Images	Classification Accuracy		Adversariality		ℓ_2
	mean	stdev	mean	stdev	mean
Original	70.0%	36.4%	0.01%	0.3%	0
Adversarial	0.9%	2.0%	96.4%	4.4%	5.6×10^{-5}



3D Adversarial Attacks

We generate 200 3D adversarial examples and evaluate in simulation.

Images	Classification Accuracy		Adversariality		ℓ_2
	mean	stdev	mean	stdev	mean
Original	68.8%	31.2%	0.01%	0.1%	0
Adversarial	1.1%	3.1%	83.4%	21.7%	5.9×10^{-3}



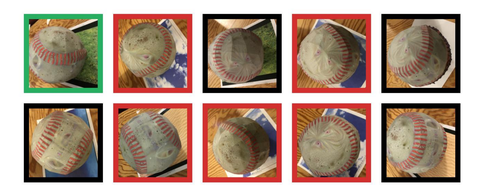
Physical World Adversarial Examples

We generate two 3D targeted adversarial examples with random target classes.

Originals are classified correctly 100% of the time:

Adversarial objects are classified as the target class most of the time:

Object	Adversarial	Misclassified	Correct
Turtle	82%	16%	2%
Baseball	59%	31%	10%



In 2018 International Conference on Machine Learning, Stockholm, Sweden