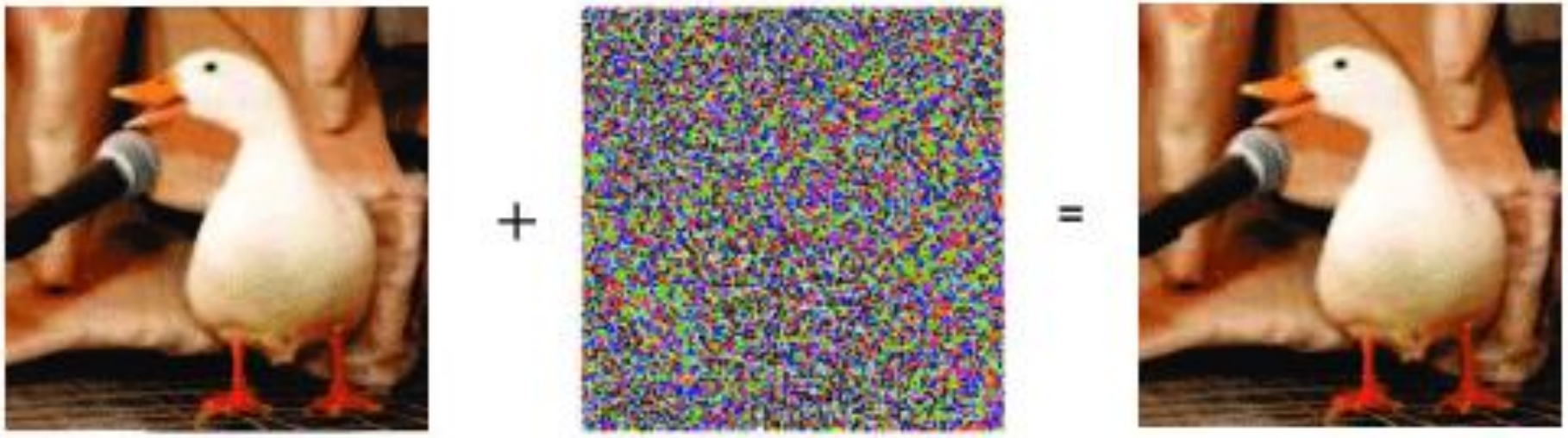
Adversarial examples in the physical world

(Kurakin, A; Goodfellow, I; Bengio, S; 2018)

Presented by: Itamar Salazar

Adversarial examples?

- Given a model M and a sample C such that $\mathrm{M}(\mathrm{C}) = \mathrm{y}_{\mathrm{true}}$
- A is an adversarial example if $\mathrm{M}(\mathrm{A}) \neq \mathrm{y}_{\mathrm{true}}$
- …and **$\mathbf{A}$ is very similar to $\mathbf{C}$**.



Adversarial examples?

- Adversarial examples are misclassified **far more often than noise**, even if the magnitude of noise is much larger (Szegedy, 2014).
- Adversarial examples **can transfer from one model to another** (Szegedy, 2014; Papernot, 2016a; Papernot, 2016b).
- Prior work has focused on constructing adversarial examples digitally, but it was not known whether **adversarial examples would remain misclassified if they were constructed in the physical world** and observed through a camera.

Can adversarial examples survive in the physical world?

- Adversarial examples exist and can be created given a model.
- But can they still be effective when viewed through a camera in the physical world?
- To find out: authors generated adversarial example, **printed them**, and tested their classification.

Related work

- **Carlini, 2016:** Created audio inputs that mobile phones recognize as intelligible voice commands but humans cannot understand.
- **Smith, 2015:** Demonstrated the vulnerability of face recognition systems to replay attacks using previously captured images of authorized users' faces.
- **Sharif, 2016:** Printed images of adversarial examples on paper and demonstrated that the printed images fool image recognition systems when photographed.
- This work differs in that they used a cheaper, simpler attack method, made no special modifications to improve the chances of survival in the physical world, and could modify all pixels.

This approach:

- Used a **pre-trained ImageNet Inception model** (M) and generated adversarial examples (A) for M, which were then fed to M through a cellphone camera to measure classification accuracy.
- A large fraction of adversarial examples remained misclassified.
- Did not make **any changes to the attack methodology** and this provides a lower bound on the attack success rate.
- Assumed a threat model under which the **attacker has full knowledge of the model architecture and parameter values**.
- Explored how adversarial examples transfer across several specific kinds of **synthetic image transformations**

Rest of the presentation

- Methods for generating adversarial images
- Photos of adversarial examples
- Artificial image transformations
- Conclusions

Methods for generating adversarial images

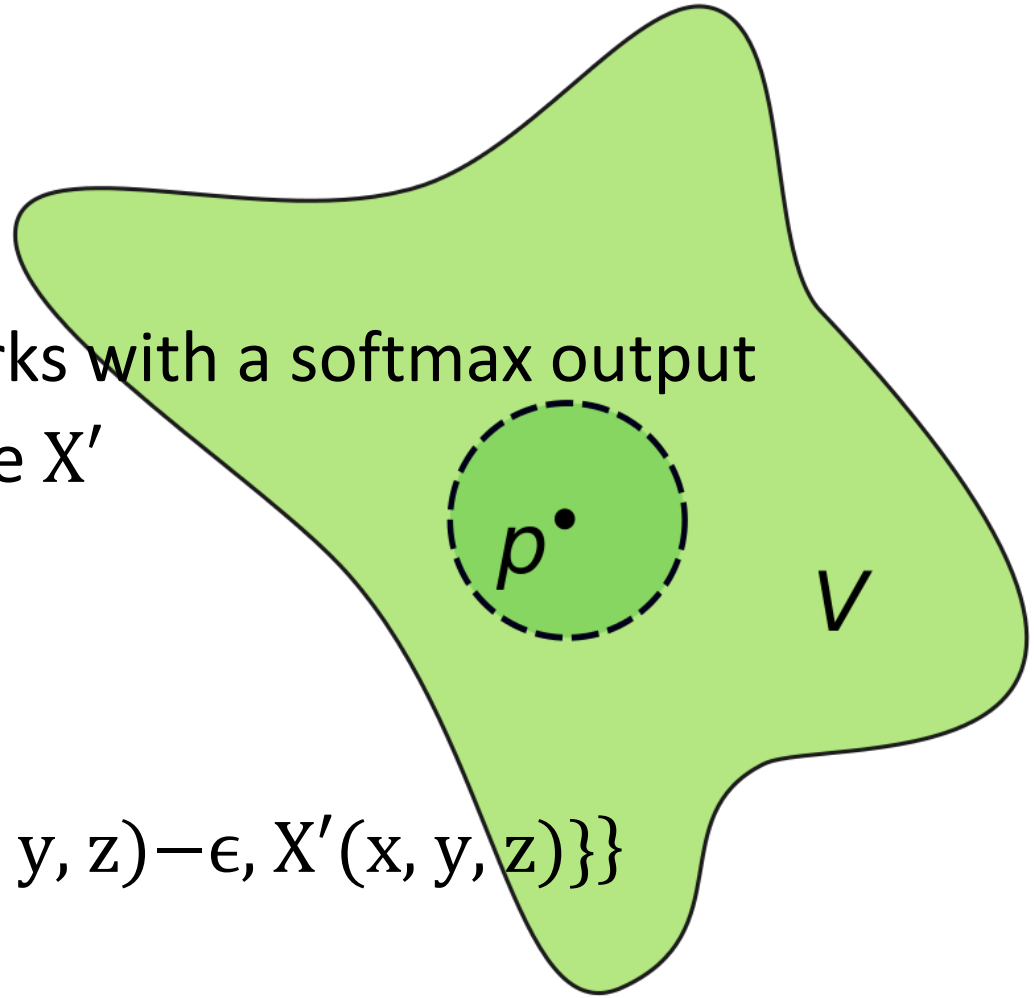
- X: image
- $\mathrm{y}_{\mathrm{true}}$: true class of X
- $\mathrm{J}(\mathrm{X}, \mathrm{y})$: cross-entropy
- $\mathrm{J}(\mathrm{X}, \mathrm{y}) = -\log \mathrm{p}(\mathrm{y}|\mathrm{X})$, for neural networks with a softmax output
- $\mathrm{Clip}_{\mathrm{X},\epsilon}\{\mathrm{X}'\}$: per-pixel clipping of the image X'

The idea is: the result will be in

L_∞ $\epsilon -$ neighbourhood of the source image

- $\mathrm{Clip}_{\mathrm{X},\epsilon}\{\mathrm{X}'\}(x, y, z)$

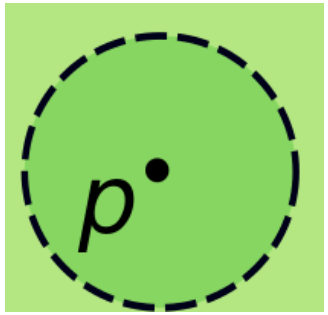
$$= \min\{255, \mathrm{X}(\mathrm{x}, \mathrm{y}, \mathrm{z}) + \epsilon, \max\{0, \mathrm{X}(\mathrm{x}, \mathrm{y}, \mathrm{z}) - \epsilon, \mathrm{X}'(\mathrm{x}, \mathrm{y}, \mathrm{z})\}\}$$



Methods for generating adversarial images

- **Fast method:**

$$X_{adv} = X + \epsilon \times \text{sign}\left(\nabla_X J(X, y_{true})\right)$$



- **Basic iterative method:**

$$X_0^{adv} = X$$

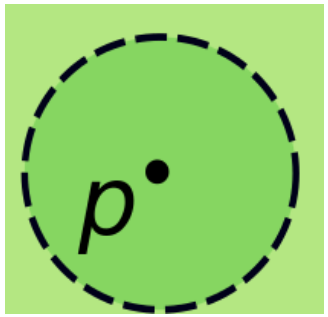
$$X_{N+1}^{adv} = Clip_{X,\epsilon}\left\{X_N^{adv} + \alpha \times \text{sign}\left(\nabla_X J\left(X_N^{adv}, y_{true}\right)\right)\right\}$$

- **Iterative least-likely class method:**
- For desired class we chose the least-likely class according to the prediction of the trained network on image X

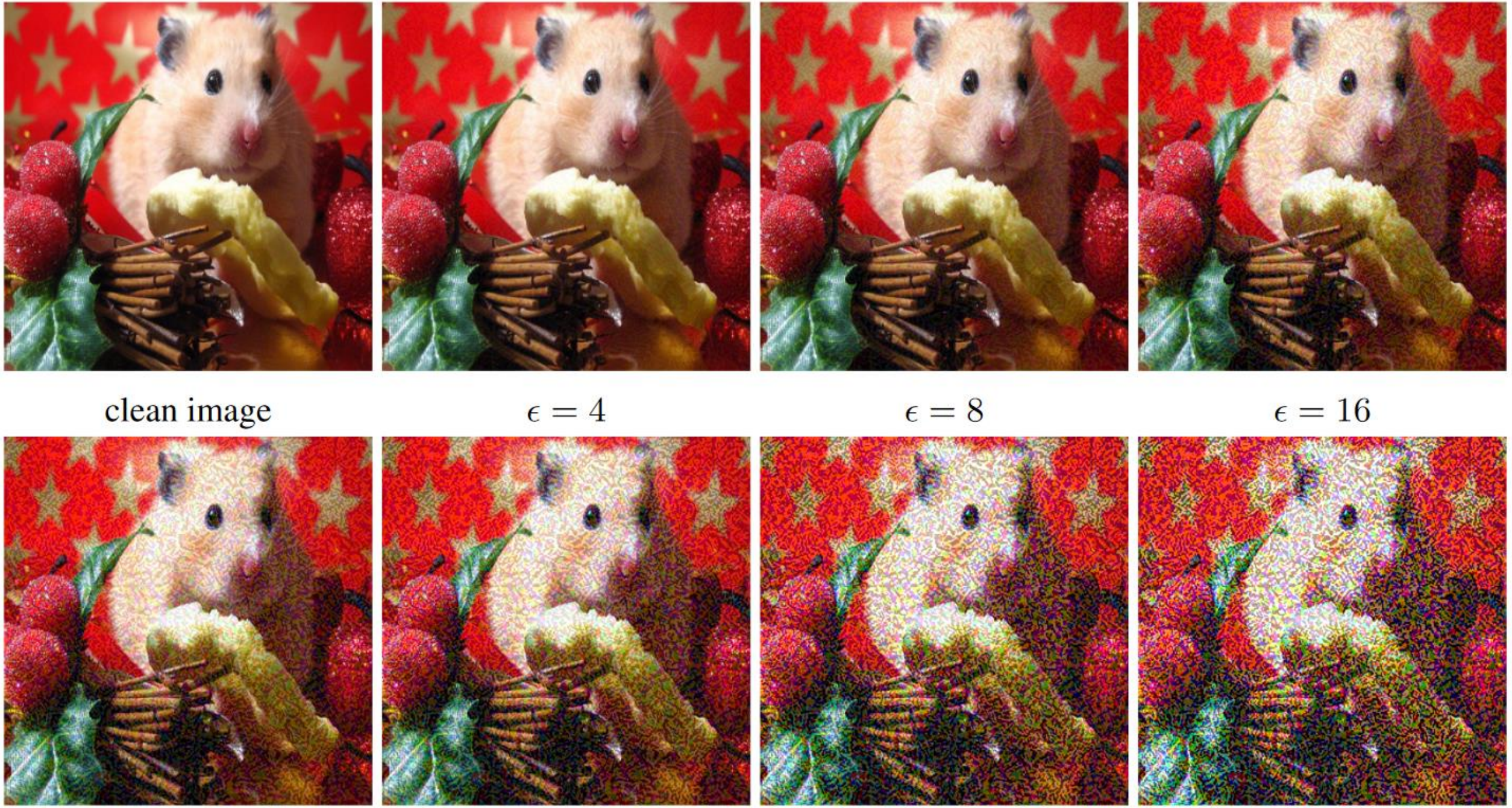
$$y_{LL} = \underset{y}{\text{argmin}}\{p(y|X)\}$$

$$X_0^{adv} = X$$

$$X_{N+1}^{adv} = Clip_{X,\epsilon}\left\{X_N^{adv} - \alpha\, sign\left(\nabla_X J\left(X_N^{adv}, y_{LL}\right)\right)\right\}$$




clean image
$\epsilon = 4$
$\epsilon = 8$
$\epsilon = 16$
$\epsilon = 24$
$\epsilon = 32$
$\epsilon = 48$
$\epsilon = 64$


clean image
$\epsilon = 4$
$\epsilon = 8$
$\epsilon = 16$
$\epsilon = 24$
$\epsilon = 32$
$\epsilon = 48$
$\epsilon = 64$

Clean image

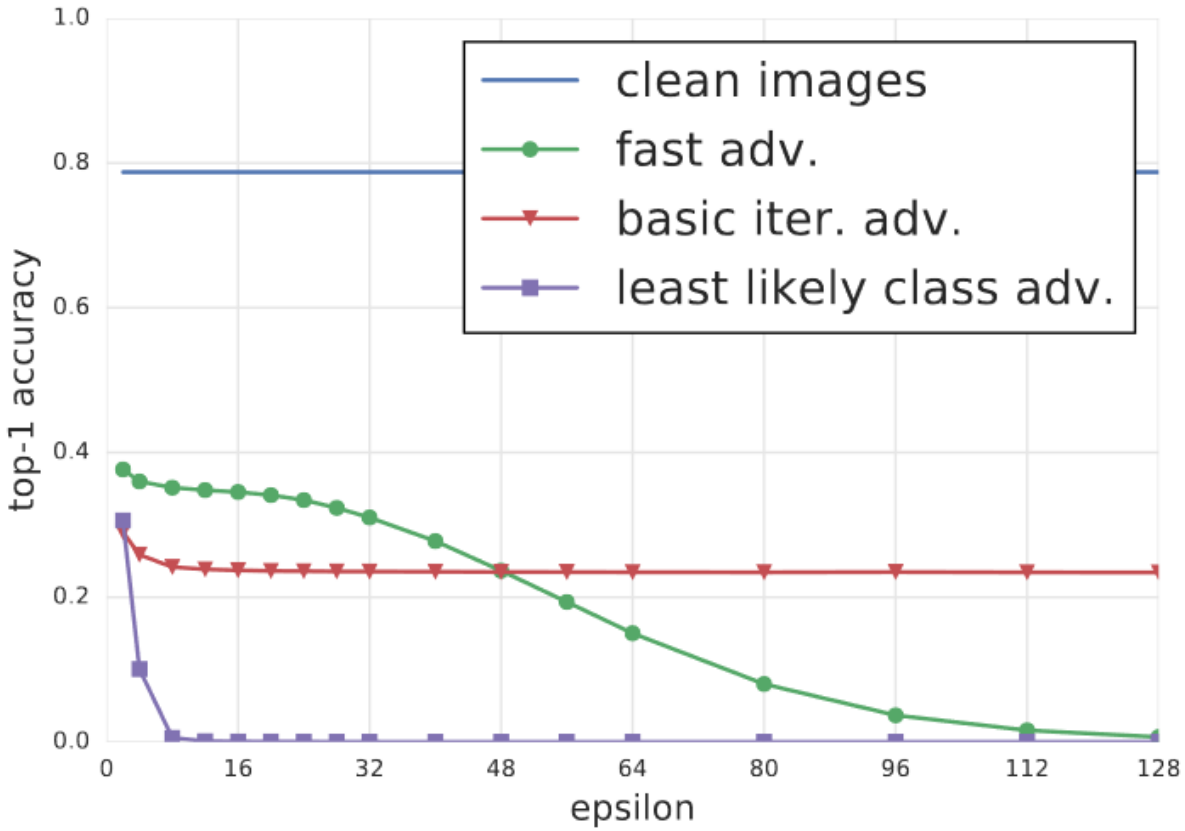
"Fast"; L_∞ distance to clean image = 32

"Basic iter."; L_∞ distance to clean image = 32

"L.l. class"; L_∞ distance to clean image = 28

Methods for generating adversarial images

- Comparison between methods:
  - All 50, 000 validation samples from the ImageNet dataset
  - Used a pre-trained Inception v3 classifier



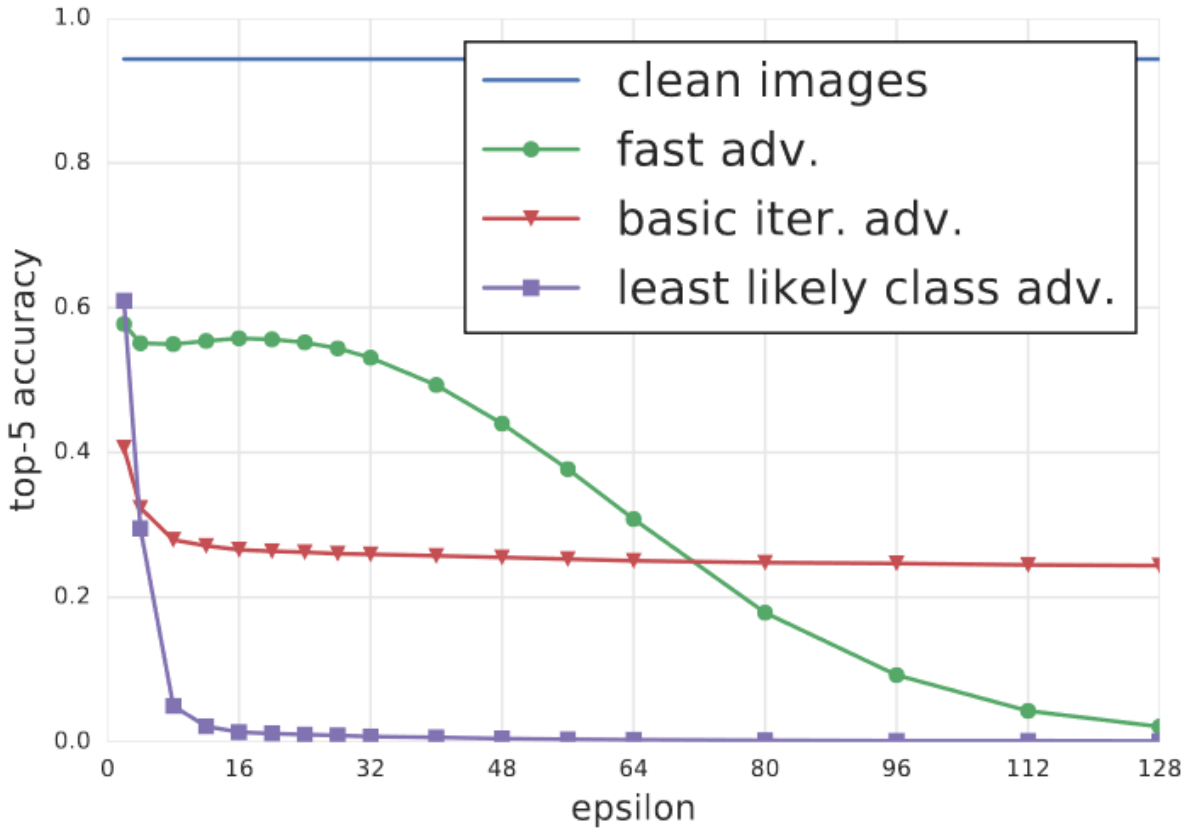


Figure 2: Top-1 and top-5 accuracy of Inception v3 under attack by different adversarial methods and different ϵ compared to "clean images" — unmodified images from the dataset. The accuracy was computed on all $50,000$ validation images from the ImageNet dataset. In these experiments ϵ varies from 2 to 128.

Further experiments are done with $\epsilon \leq 16$

Photos of adversarial examples

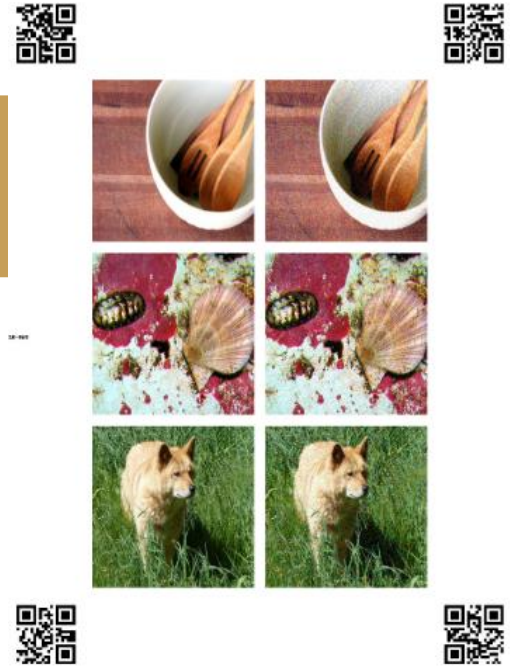
- **Destruction Rate of Adversarial Images:** The fraction of adversarial images that are no longer misclassified after transformations

$$d = \frac{\sum_{k=1}^{n} C(\boldsymbol{X}^k, y_{true}^k)\overline{C(\boldsymbol{X}_{adv}^k, y_{true}^k)}C(T(\boldsymbol{X}_{adv}^k), y_{true}^k)}{\sum_{k=1}^{n} C(\boldsymbol{X}^k, y_{true}^k)\overline{C(\boldsymbol{X}_{adv}^k, y_{true}^k)}}$$

- Function T(•) represents arbitrary image transformations
- Study of various transformations:
  - One transformation: Printing the image and taking a photo of the result.

Experimental setup

- **Average Case**
  - Randomly selected 102 images
- **Prefiltered Case**
  - Selected 102 images based on classification accuracy:
  - All clean images are classified correctly
  - All adversarial images (before photo transformation) are misclassified (both top-1 and top-5 classification)



(a) Printout

(b) Photo of printout

(c) Cropped image

Table 1: Accuracy on photos of adversarial images in the average case (randomly chosen images).

Adversarial method	Photos				Source images			
	Clean images		Adv. images		Clean images		Adv. images	
	top-1	top-5	top-1	top-5	top-1	top-5	top-1	top-5
fast $\epsilon = 16$	79.8%	91.9%	36.4%	67.7%	85.3%	94.1%	36.3%	58.8%
fast $\epsilon = 8$	70.6%	93.1%	49.0%	73.5%	77.5%	97.1%	30.4%	57.8%
fast $\epsilon = 4$	72.5%	90.2%	52.9%	79.4%	77.5%	94.1%	33.3%	51.0%
fast $\epsilon = 2$	65.7%	85.9%	54.5%	78.8%	71.6%	93.1%	35.3%	53.9%
iter. basic $\epsilon = 16$	72.9%	89.6%	49.0%	75.0%	81.4%	95.1%	28.4%	31.4%
iter. basic $\epsilon = 8$	72.5%	93.1%	51.0%	87.3%	73.5%	93.1%	26.5%	31.4%
iter. basic $\epsilon = 4$	63.7%	87.3%	48.0%	80.4%	74.5%	92.2%	12.7%	24.5%
iter. basic $\epsilon = 2$	70.7%	87.9%	62.6%	86.9%	74.5%	96.1%	28.4%	41.2%
l.l. class $\epsilon = 16$	71.1%	90.0%	60.0%	83.3%	79.4%	96.1%	1.0%	1.0%
l.l. class $\epsilon = 8$	76.5%	94.1%	69.6%	92.2%	78.4%	98.0%	0.0%	6.9%
l.l. class $\epsilon = 4$	76.8%	86.9%	75.8%	85.9%	80.4%	90.2%	9.8%	24.5%
l.l. class $\epsilon = 2$	71.6%	87.3%	68.6%	89.2%	75.5%	92.2%	20.6%	44.1%

Table 2: Accuracy on photos of adversarial images in the prefiltered case (clean image correctly classified, adversarial image confidently incorrectly classified in digital form being being printed and photographed).

Adversarial method	Photos				Source images			
	Clean images		Adv. images		Clean images		Adv. images	
	top-1	top-5	top-1	top-5	top-1	top-5	top-1	top-5
fast $\epsilon = 16$	81.8%	97.0%	5.1%	39.4%	100.0%	100.0%	0.0%	0.0%
fast $\epsilon = 8$	77.1%	95.8%	14.6%	70.8%	100.0%	100.0%	0.0%	0.0%
fast $\epsilon = 4$	81.4%	100.0%	32.4%	91.2%	100.0%	100.0%	0.0%	0.0%
fast $\epsilon = 2$	88.9%	99.0%	49.5%	91.9%	100.0%	100.0%	0.0%	0.0%
iter. basic $\epsilon = 16$	93.3%	97.8%	60.0%	87.8%	100.0%	100.0%	0.0%	0.0%
iter. basic $\epsilon = 8$	89.2%	98.0%	64.7%	91.2%	100.0%	100.0%	0.0%	0.0%
iter. basic $\epsilon = 4$	92.2%	97.1%	77.5%	94.1%	100.0%	100.0%	0.0%	0.0%
iter. basic $\epsilon = 2$	93.9%	97.0%	80.8%	97.0%	100.0%	100.0%	0.0%	1.0%
l.l. class $\epsilon = 16$	95.8%	100.0%	87.5%	97.9%	100.0%	100.0%	0.0%	0.0%
l.l. class $\epsilon = 8$	96.0%	100.0%	88.9%	97.0%	100.0%	100.0%	0.0%	0.0%
l.l. class $\epsilon = 4$	93.9%	100.0%	91.9%	98.0%	100.0%	100.0%	0.0%	0.0%
l.l. class $\epsilon = 2$	92.2%	99.0%	93.1%	98.0%	100.0%	100.0%	0.0%	0.0%

Table 3: Adversarial image destruction rate with photos.

Adversarial method	Average case		Prefiltered case	
	top-1	top-5	top-1	top-5
fast $\epsilon = 16$	12.5%	40.0%	5.1%	39.4%
fast $\epsilon = 8$	33.3%	40.0%	14.6%	70.8%
fast $\epsilon = 4$	46.7%	65.9%	32.4%	91.2%
fast $\epsilon = 2$	61.1%	63.2%	49.5%	91.9%
iter. basic $\epsilon = 16$	40.4%	69.4%	60.0%	87.8%
iter. basic $\epsilon = 8$	52.1%	90.5%	64.7%	91.2%
iter. basic $\epsilon = 4$	52.4%	82.6%	77.5%	94.1%
iter. basic $\epsilon = 2$	71.7%	81.5%	80.8%	96.9%
l.l. class $\epsilon = 16$	72.2%	85.1%	87.5%	97.9%
l.l. class $\epsilon = 8$	86.3%	94.6%	88.9%	97.0%
l.l. class $\epsilon = 4$	90.3%	93.9%	91.9%	98.0%
l.l. class $\epsilon = 2$	82.1%	93.9%	93.1%	98.0%

Experimental results on photos of adversarial examples

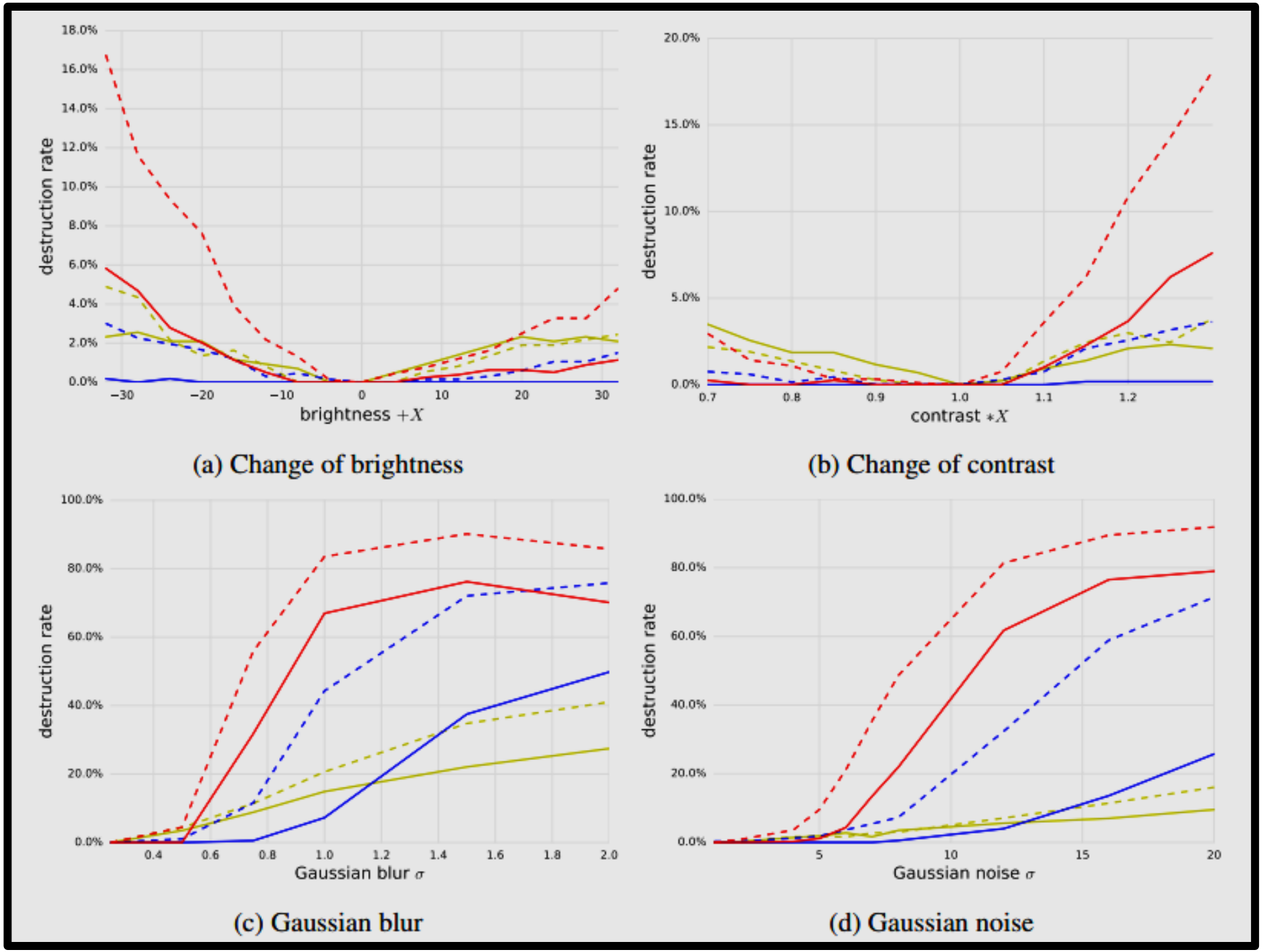
- Robustness of Adversarial Images to Photo Transformation
  - "Fast" adversarial images are more robust compared to iterative methods
  - Iterative methods exploit more subtle perturbations
- Adversarial Destruction Rate in Different Cases
  - "Prefiltered case" had higher destruction rate in some cases compared to "average case"
  - Iterative methods make subtle co-adaptations for high confidence that cannot survive photo transformation
- Conclusion
  - **Some adversarial examples stay misclassified even after photo transformation**
  - **Photo transformation is a non-trivial transformation that can impact adversarial robustness**

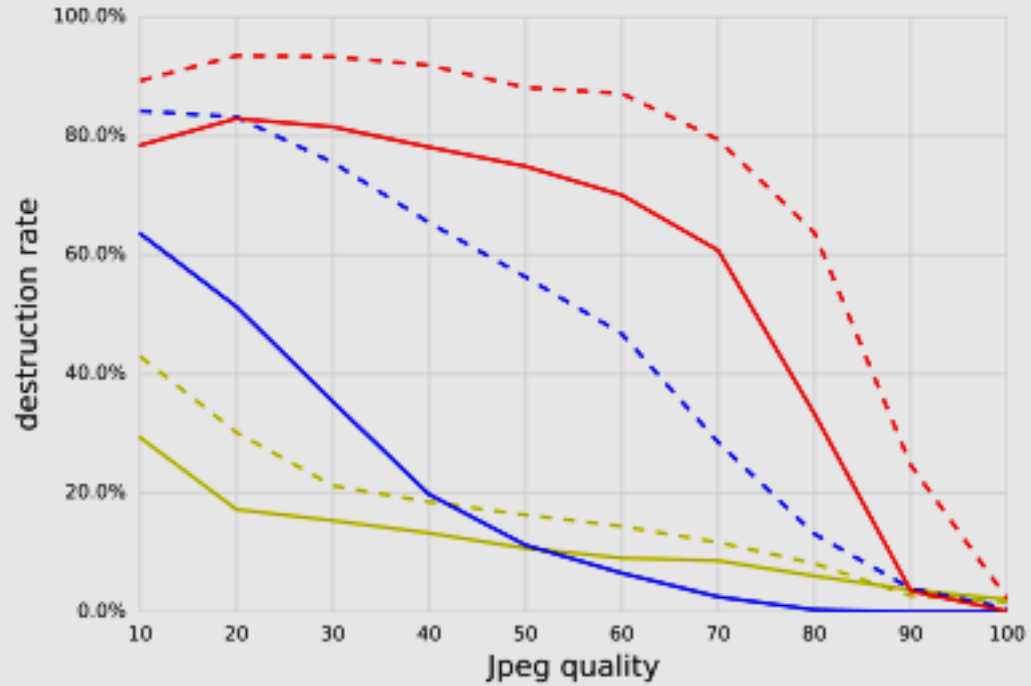
Demonstration of Black Box Adversarial Attack in the Physical World

- Fooling a different model than the one used to construct them
  - Specifically, open source TensorFlow camera demo was fooled
- Video Demonstration: https://youtu.be/zQ_uMenoBCk

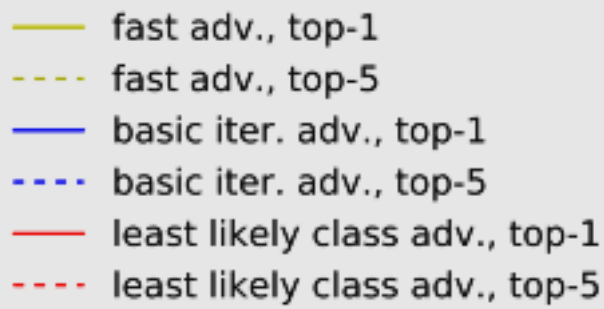
Artificial image transformations

- We explored the following set of transformations:
  - change of contrast
  - Brightness
  - Gaussian blur
  - Gaussian noise
  - JPEG encoding


destruction rate
brightness $+X$
(a) Change of brightness
destruction rate
contrast $*X$
(b) Change of contrast
destruction rate
Gaussian blur σ
(c) Gaussian blur
destruction rate
Gaussian noise σ
(d) Gaussian noise
fast adv., top-1
fast adv., top-5
basic iter. adv., top-1
basic iter. adv., top-5
least likely class adv., top-1
least likely class adv., top-5



(e) JPEG encoding



Artificial image transformations

- General Observations from Experiments:
  - **Fast** method generates the **most robust** adversarial examples to transformations
  - **Iterative least-likely** class method generates the **least robust** adversarial examples
  - Results coincide with photo transformation experiment
  - Top-5 destruction rate is typically higher than top-1 destruction rate
  - Pushing correct class labels into top-5 predictions is easier
- Brightness and Contrast Transformations
  - **Do not affect adversarial examples much**
  - Destruction rate for fast and basic iterative adversarial examples is less than 5%
  - Destruction rate for iterative least-likely class method is less than 20%
- Blur, Noise, and JPEG Encoding Transformations
  - **Have a higher destruction rate compared to brightness and contrast transformations**
  - Destruction rate for iterative methods could reach 80% - 90%
- None of these transformations destroy 100% of adversarial examples, coinciding with photo transformation experiment.

Conclusions

- Findings
  - Significant fraction of adversarial images crafted using the original network are misclassified even when fed to the classifier through a cell-phone camera
  - **Demonstrates possibility of adversarial examples for machine learning systems in the physical world**
- Future Work
  - Demonstrate attacks using other physical objects besides printed images
  - Attacks against different machine learning systems, such as reinforcement learning agents
  - Attacks performed without access to model's parameters and architecture
  - Physical attacks achieving higher success rates by modeling physical transformation during construction process

Thanks

Presented by: Itamar Salazar