



#### Introduction

In this work, we explore adversarial attacks by visualizing class activation mapping (CAM) [1] and graphical saliency [2] as two feature attribution techniques.

We also compare feature visualization techniques across various image classifiers to exemplify the phenomena of adversarial examples and investigate their training as a structural regularizer. We experiment with a range of epsilon values ( $\epsilon$ ) as our independent variable, and compare the results under ImageNet images and different model configurations. We also introduce AdVis.js, an interactive system for generating adversarial examples and explaining them for the first time in real-time.





57.7% confidence

 $sign(\nabla_x J(\theta, x, y))$ 

8.2% confidence



 $\epsilon sign(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}))$ 99.3 % confidence

Figure 1. Fast Gradient Sign Method applied to an nage of a panda causes he model to misclassify he image as a 'gibbon' with a high confidence. Image Source: [1].

Adversarial attacks are performed by perturbing the input image to a classifier such that it misclassifies the input with high confidence while the modification is imperceptible to humans. A key parameter to canonical gradient-based attack named Fast Gradient Sign method (FGSM) is epsilon, which determines the amount of perturbation applied.

<b>AdVis</b>	s.js									
		jlin.xyz/	advis/	Ċ	ثل ا					
■ AdVis.js   Visualizing Adv	versarial Attacks									
		AdV	is.js							
		Exploring Fa	st Gradient							
		Sign ivi	elnoa							
		Explore Adversarial Attacks w	ith visual interactive tools.							
How can we detect an adversarial example? When you see a corrupted image of, let's say, a panda - you recognize it. Probably by the colorful noise. But for the machine it's not a noisy photo of a panda, it's a chihuahua. And it's so sure about it, that it doesn't make sense to question its own decisions. AdVis.js lets you explore adversarial attacks by dynamically presenting the classification scores and CAM heatmap visualization as you tune the strength of perturbation applied in real-time. The changing value via the solider below and see for yourself!										
Upload your input image Choose Files of files selected										
<b>Advis.js</b> is the first to bring visual adversarial example generation and dynamic visualization to the browser for real-time exploration, and we invite developers and researchers to contribute to our growing library of attack vectors.										
	Class	Confidence %		Class	Confidence %					
click rows to toggle heatmap →	giant panda, panda, panda bear, coon bear, Ailuropoda melanoleuca	99.07	1254	Chihuahua	93.87					
	indri, indris, Indri indri, Indri brevicaudatus	0.51		Pomeranian	1.61					
	gibbon, Hylobates lar	0.08	<b>NORMAN</b>	toy poodle	1.10					
	lesser panda, red panda, panda, bear cat, cat bear, Ailurus fulgens	0.05		miniature poodle	0.47					
	titi, titi monkey	0.04	Epsilon: 13.06	mongoose	0.44					
+			NEW ATTACK!							
		- 1000 Bothon Movimization	_	_						

Figure 2. AdVis landing page





A real-time web application built with Tensorflow.js and React, AdVis.js lets users explore adversarial attacks by dynamically updating classification scores to reflect different epsilon values used to generate attacks with minimal latency. Users can also visualize the CAM overlay for a specific class on demand as they continue to tune their desired epsilon values. AdVis currently supports Fast Gradient Sign Method on MobileNet, with additional gradient and deep learning based targeted attacks, as well as classifier models to be added in the future.

# AdVis: Visualizing and Attributing ML Attacks to Adversarial Examples

Jason Lin, Dilara Soylu Advisor: Dr. James Hays, Georgia Tech

### **Visualization Experiments**

1. Effects of changing epsilon on CAM and dissimilarity-based saliency detection Below demonstrates how a range of epsilon values used for the FGSM attack alters the CAM of the target class and saliency maps on the original images. From the CAM visualization we see that as epsilon ( $\epsilon$ ) increases, more pixels strongly contribute to the activation of the class, and the yellow area (row 2) appears more illuminated and geometrically shifted. The change in saliency heatmap overlay (row 3) as we vary  $\epsilon$ does not appear strikingly different to the human eye, although the saliency difference grows significantly, which is illustrated in the next section.



**2.** Comparing saliency difference between original and perturbed image pairs We define saliency difference as the saliency map of original image subtracted from that of the perturbed image. Below images show how the saliency map difference varies as epsilon increases. Interestingly, we see a positive correlation between the amount of perturbation specified by epsilon and the area of graphical saliency, which is proportional to pixel-value dissimilarity and proximity from neighborhood pixels.  $w_1((i,j),(p,q)) \triangleq d((i,j)||(p,q)) \cdot F(i-p,j-q)$  (2)

Because epsilon scales the factor of gradient-derived filter applied, we speculate that the above graphical dissimilarity metric (2), independent of any labels, encodes perturbations conditioned on the target class as saliency. We discover a novel correlation between log-ratio saliency difference and ConvNet-inspired CAM.



[3] states that gradient-based attacks are transferable across architectures. However, we found that FGSM is not as nearly effective when the target model is different from the attack model. Below, we study how the adversarial examples generated with MobileNetV1 perform on MobileNetV1 vs. MobileNetV2 (white-box vs black-box attack). The classification results do not update as desired for MobileNetV2, until epsilon becomes sufficiently large. Perhaps the inverted residual connections introduced in MobileNetV2 [4] to address the vanishing gradients problem with depthwise convolutions can explain this.

classi adver image **Mobil** 

Mobile classifi adver images

**3.** Comparison of the saliency difference between the original and the perturbed images for different models and epsilon values The attacks are more successful as epsilon increases, which is proportional to the saliency map difference. Despite attacks on MobilenetV1 being more effective, the saliency map difference seems smaller than that obtained for attacks on MobilenetV2. We suspect that this is because as MobileNetV2 tries to retain more gradients, it corresponds directly to the greater saliency size we hypothesized in section 2 and suggests that gradients may not be the primary indicator of classification correctness, where more concentrated perturbations appear more effective. ε = 10

Saliency detection difference between the *Panda* image and the adversarial images generated with **MobilenetV1** 

Saliency detection difference betwee the Panda image and the adversarial images generated with **MobilenetV2** 

Saliency detection difference between adversarial Panda images generated with Inception and MobilenetV1





#### **Transferability across Networks**

	ε = 1	ε = 4	ε = 7	ε = 10	ε = 13	ε = 16
lenetV1 ification for the rsarial <i>Panda</i> es generated with <b>lenetV1</b>	Giant panda: 3.47 Howler monkey: 4.07 Indri: 2.10 Gibbon: 1.61 Black-footed ferret:1.58	Giant panda: 7.05 Bow-tie: 5.68 Cocker spaniel: 4.68 Toy poodle: 4.30 Chihuahua: 3.67	Cocker spaniel: 18.57 Toy poodle: 10.88 bow-tie : 5.14 Beagle: 3.91	Cocker spaniel: 25.25 Toy poodle: 9.54 Chihuahua: 9.33 Beagle: 4.33 Coach dog: 3.11	Chihuahua: 26.53 Cocker spaniel: 25.84 Toy poodle: 7.17 Beagle: 4.42 Giant panda: 3.20	Chihuahua: 31.40 Cocker spaniel: 22.87 Giant panda: 5.09 Toy poodle: 4.76 West highland white terrier: 4.30
enetV1 fication for the sarial <i>Panda</i> is generated with <b>enetV2</b>	Giant panda: 67.49 Gibbon: 6.66 Howler monkey: 6.44 Indri: 4.02 Capuchin: 1.34	Giant panda: 85.11 Gibbon: 4.67 Iangur: 1.73 Howler monkey: 1.03 Chihuahua: 0.91	Giant panda: 62.38 Gibbon: 7.79 chihuahua: 5.66 Lesser panda: 2.53 Pomeranian: 2.07	Giant panda: 35.29 Chihuahua: 13.64 Siamese cat: 10.31 Pomeranian: 6.90 Red panda: 3.76	Siamese cat: 30.32 chihuahua: 12.90 Giant panda: 12.63 pomeranian: 9.09 West highland white terrier: 6.90	Siamese cat: 30.16 chihuahua: 18.71 pomeranian: 12.25 West highland white terrier: 6.34 Giant panda: 4.2



#### **Feature Visualization vs. Attribution**





SqueezeNet - CAM

**MobileNet - CAM** 

**Graphical Saliency** 

DeepDream

A study of CAM on different models vs. feature visualization methods like DeepDream exposes intra-and interclass similarities between the two kinds of interpretability.

#### **Data Augmentation**

4. MobileNet classification of downsized vs. center-cropped images perturbed with GoogLeNet

adversarially generated at 299x299 with Inception and center cropped to 224x224

adversarially generated at 299x299 with nception and lownsized t 224x224

A discovery found comparing MobileNetV1 with GoogLeNet is that fringe/edge areas may be critical to an attack's effectiveness. Using the correlation between MobileNet classification and graphical saliency, which tends to focus on the center of image, we speculate that to preserve what humans perceive, the FGSM perturbations implicitly gravitate towards borders of the image, where after upscaling the image from 224 x224 to 299x299 to fit Inception's requirements and then center-cropping to 224x224 for MobileNet classification, the results are indeed highly sensitive and susceptible to change in the outer regions. MobileNet seems to be more easily confused with a consistent classification across  $\epsilon$  values when it is cropped to the center.

## **Comparing Attack Vectors**

5. Experimentation with DeepFool: To explore the relation between gradients and robustness of adversarial attacks, we refer to another attack vector DeepFool [5], which cites [3] as an efficient but coase approximation of suboptimal perturbation vectors. DeepFool is just as effective as FGSM, although its salient region almost seem as an inverted version of that of FGSM's (fig. 3 vs 4). This suggests that the two methods are modifying different parts of the image to render the adversarial versions of input images.



#### demo, code, and data at http://github.com/jaxball/advis.js References

College of Computing

	ε = 1		ε = 4		ε=7		€ = 10		ε = 13		ε = 16	
	Class	Confidence %	Class	Confidence %	Class	Confidence %	Class	Confidence %	Class	Confidence %	Class	Confidence %
	colobus, colobus monkey	24.85	gibbon, Hylobates lar	21.66	Chihuahua	22.49	Chihuahua	22.63	Chihuahua	23.13	Chihuahua	23.11
	giant panda, panda, panda bear, coon bear	13.31	squirrei monkey, Saimiri sciureus	11.90	llama	12.48	llama	12.61	llama	12.18	llama	12.26
	Ailuropoda melanoleuca		howler monkey, howler	10.20	vulture	5.66	vulture	5.60	vulture	5.62	vulture	5.62
	gibbon, Hylobates lar	6.66	giant panda, panda, panda bear,	7 44	puffer, pufferfish		puffer, pufferfish.		puffer, pufferfish,		gibbon, Hylobates lar	3.11
	meerkat, mierkat	5.39	coon bear, Ailuropoda melanoleuca	7.44	blowfish, globefish	3.20	blowfish, globefish	h, 3.16 sh	blowfish, globefish	3.13	puffer,	
1922	howler monkey, howler	4.71	indri, indris, Indri indri, Indri brevicaudatu	6.54	gibbon, Hylobates lar	3.12	gibbon, Hylobates lar	3.14	gibbon, Hylobates lar	3.04	blowfish, globefish	3.06
	Class	Confidence %	Class	Confidence %	Class	Confidence %	Class	Confidence %	Class	Confidence %	Class	Confidence %
	gibbon, Hylobates lar	22.19	gibbon, Hylobates lar	21.66	gibbon, Hylobates lar	21.49	gibbon, Hylobates lar	21.49	gibbon, Hylobates lar	21.62	gibbon, Hylobates lar	21.78
	giant panda, panda, panda bear, coon bear,	17.35	squirrel monkey, Saimiri sciureus	11.90	squirrel monkey, Saimiri sciureus	11.78	squirrel monkey, Saimiri sciureus	11.88	squirrel monkey, Saimiri sciureus	12.03	squirrel monkey, Saimiri sciureus	11.93
	Alluropoda melanoleuca indri, indris,		howler monkey, howler	10.20	howler monkey, howler	10.17	howler monkey, howler	10.19	howler monkey, howler	10.54	howler monkey, howler	10.50
R18 (20)	Indri Indri, Indri brevicaudatu s	11.59	giant panda, panda, panda bear, coon bear,	7.44	giant panda, panda, panda bear, coon bear,	7.45	giant panda, panda, panda bear, coon bear,	7.26	giant panda, panda, panda bear, coon bear	7.17	giant panda, panda, panda bear, coon bear.	7.26
	howler monkey, howler	7.35	Ailuropoda melanoleuca		Ailuropoda melanoleuca indri, indris		Ailuropoda melanoleuca		Ailuropoda melanoleuca		Ailuropoda melanoleuca	
	colobus, colobus monkey	6.63	Indri, Indris, Indri indri, Indri brevicaudatu	6.54	Indri indri, Indri brevicaudatu	6.59	indri, indris, Indri indri, Indri brevicaudatu	6.55	indri, indris, Indri indri, Indri brevicaudatu	6.49	Indri, Indris, Indri indri, Indri brevicaudatu	6.40









[1] Zhou, Bolei, et al. "Learning deep features for discriminative localization." Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on. IEEE, 2016.

[2] Harel, Jonathan, Christof Koch, and Pietro Perona. "Graph-based visual saliency." Advances in neural information processing systems. 2007.

[3] Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." arXiv preprint arXiv:1412.6572 (2014).

[4] Sandler, Mark, et al. "Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation." arXiv preprint arXiv:1801.04381 (2018).

[5] Moosavi Dezfooli, Seyed Mohsen, Alhussein Fawzi, and Pascal Frossard. "Deepfool: a simple and accurate method to fool deep neural networks." Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016.