UNIVERSITÉ DE NEUCHÂTEL

UNIVERSITÄT BERN

UNIVERSITÉ DE FRIBOURG
UNIVERSITÄT FREIBURG

MASTER IN COMPUTER SCIENCE

Adversarial attacks in CNNs

Spring 2021 : Seminar Explainable AI – Human–Computer Interaction meets Artificial Intelligence

Final Presentation – May 28, 2021

VEDASRI NAKKA

Topic：Adversarial Attacks Meets Interpretability

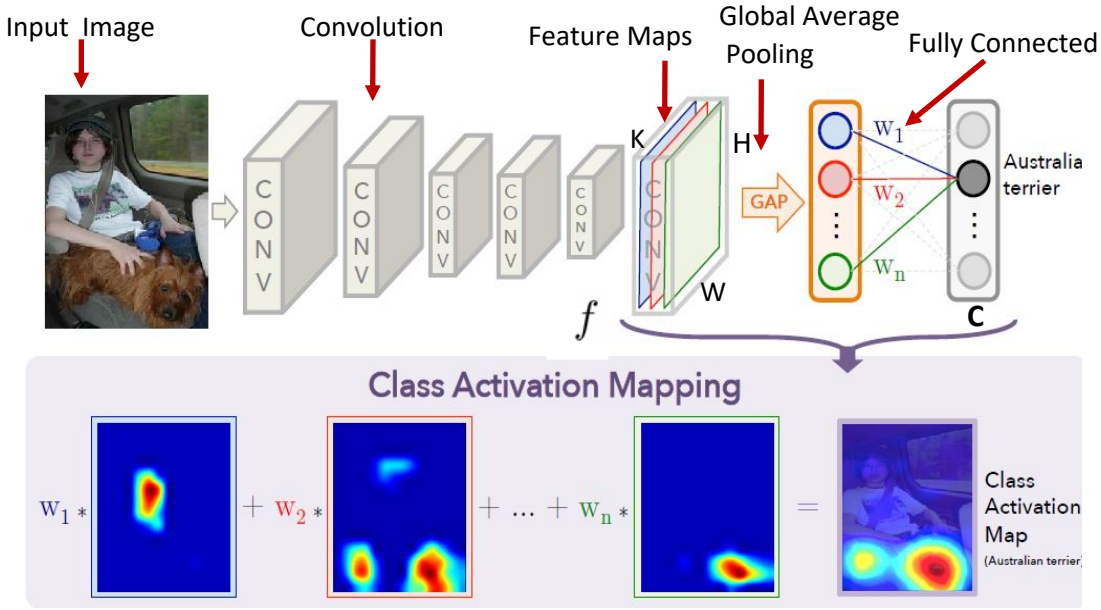
Analyze and explain the decisions of fine-grained recognition networks by studying the image regions responsible for classification for both clean and adversarial examples

Interpretability in CNNs

1. **Class Activation Maps**
2. **Attention Based Networks**
3. **Prototype Based Networks**

CAM – Deep Features for Discriminative Localization

Identify which image regions responsible for classification *after* the training



M_c : Class Activation Map of class c

$$M_c(x,y) = \sum_k w_k^c f_k(x,y)$$

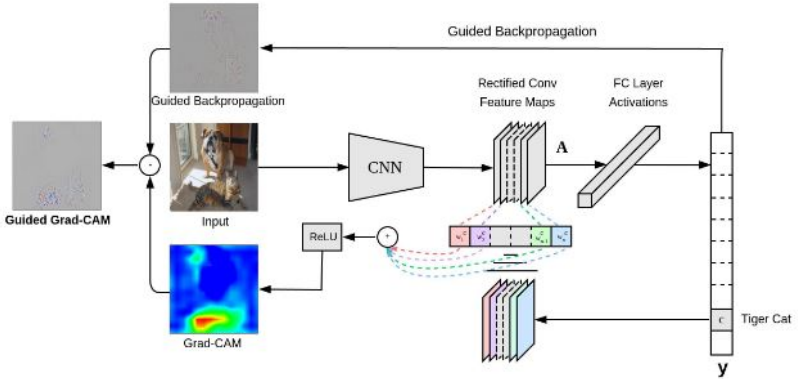
$f_k \in \mathbf{R}^{H \times W} \quad w_k \in \mathbf{R}^{K \times 1}$

Key Problem:
Limited to GAP-CNN architectures

*Bolei Zhou, et al. Learning deep features for discriminative localization, In **CVPR 2016***

Grad-CAM: Gradient-based Localization

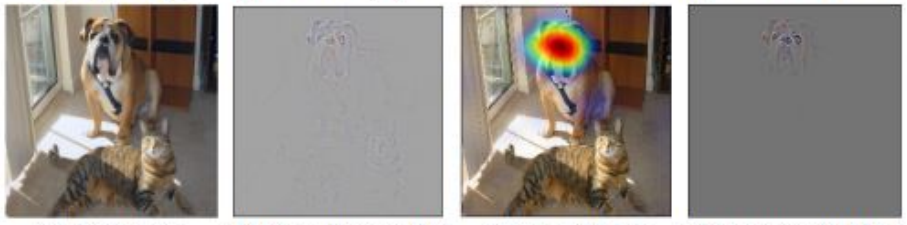
Computes the gradients instead of fixed weights of final layer



$$\alpha_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A_{ij}^k}$$

$$L_{Grad-CAM}^c = ReLU\left(\sum_k \alpha_k^c A_k^c\right)$$

Generalized to any CNN architecture

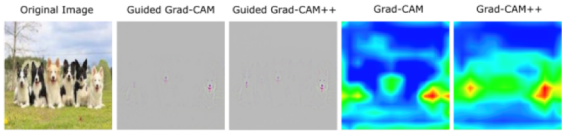


Ramprasaath R.Selvaraju, et al. Visual Explanations from Deep Networks via Gradient-based Localization, ***In ICCV 2017***

Grad CAM++

Grad–CAM++: Improved Discriminative Localization

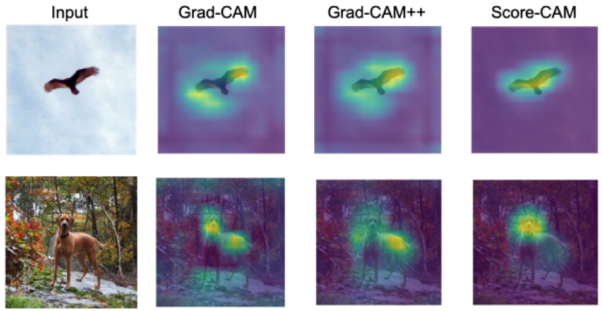
[Aditya et al., CVPR 2017]



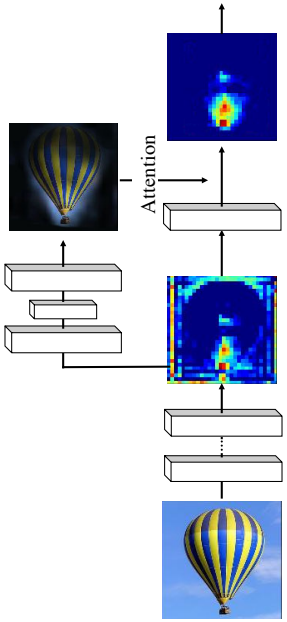
Score CAM

Score–Weighted Visual Explanations

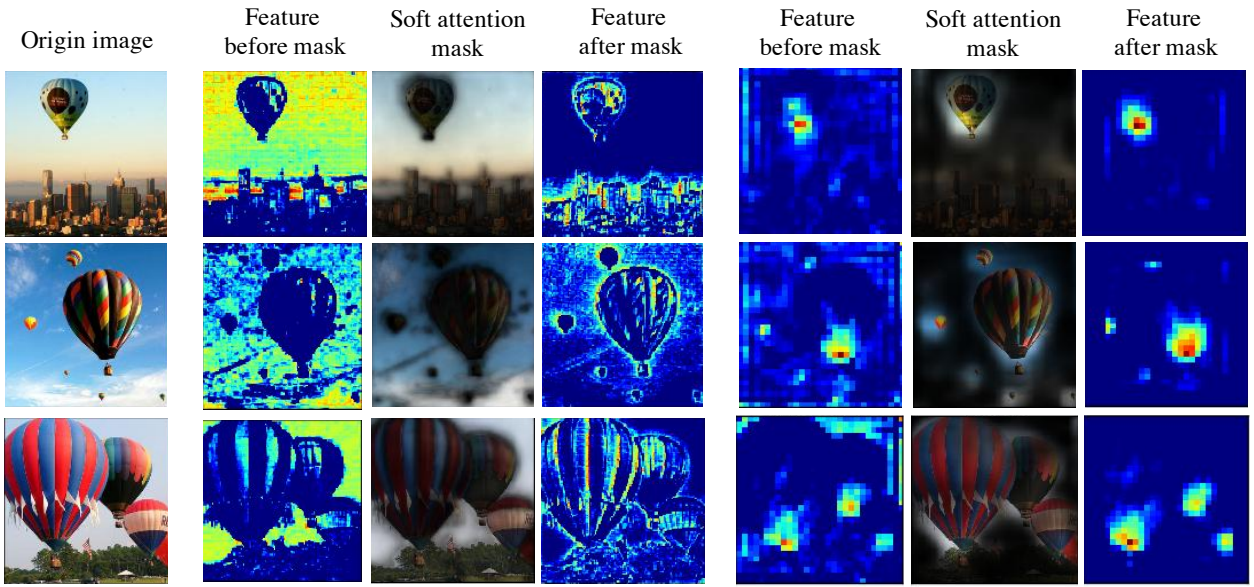
[Wang et al., CVPR 2020]



Residual Attention Network



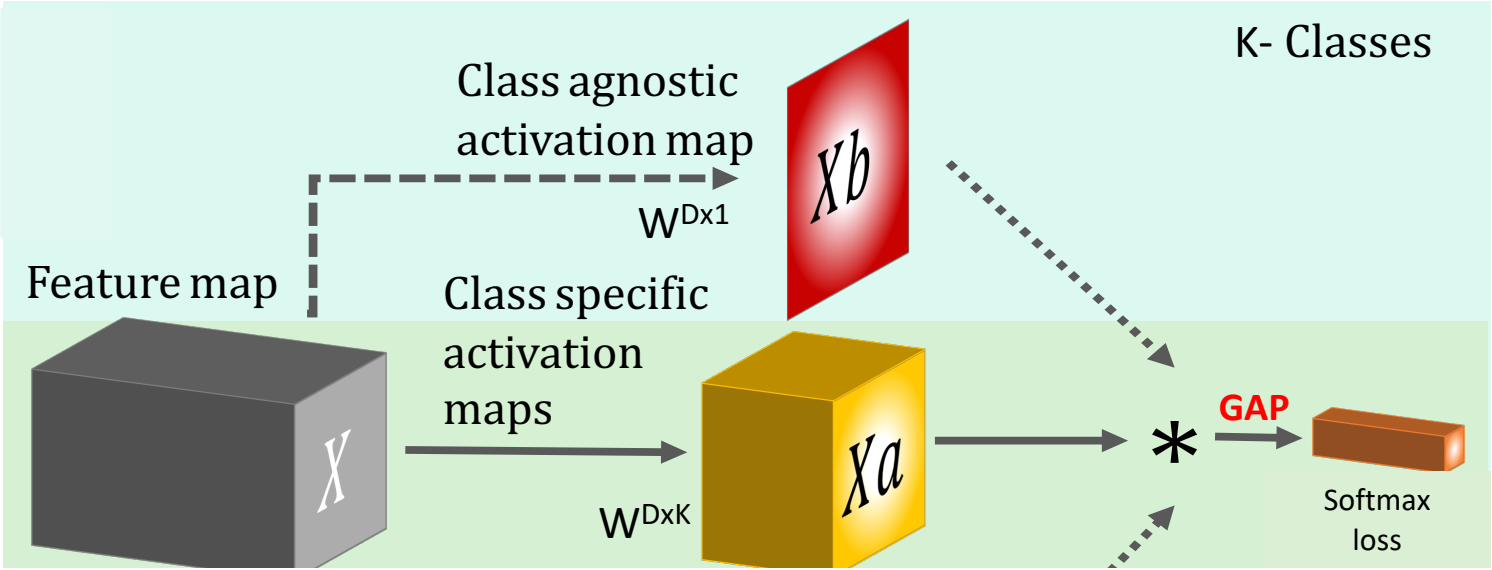
Attention mechanism



Attention mechanism incorporated with feed forward network architecture in an end-to-end training fashion

Fei Wang et al., Residual Attention Network for Image Classification, ***CVPR 2017***

Attention Pooling



CAMs = Feature maps –> GAP –> Class specific weights –> weighted feature map

Attention Pooling = Feature maps –> class specific (& agnostic) weights–> GAP

Rohit et al., Attention Pooling for Action Recognition, NIPS 2017

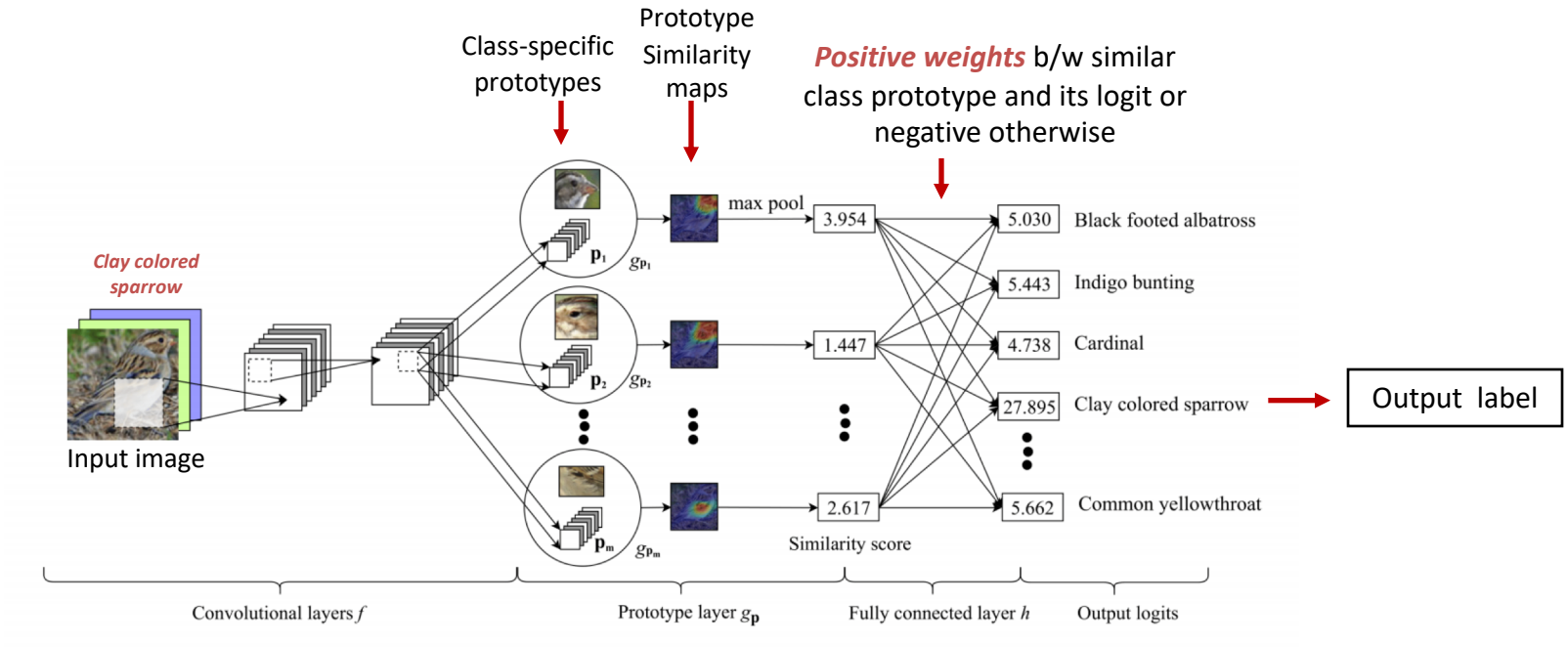
Both CAM and Attention-based models can only tell us which parts of the input they are looking at – they do not point us to prototypical cases to which the parts they focus on are similar

Solution?

Learn class–specific protoypes during training

ProtoPNet: Interpretable Fine-Grained Network

Explain the decisions of CNN through class-specific prototypes in a feed-forward manner



Chaofan Chen, et al. This Looks Like That: Deep Learning for Interpretable Image Recognition, ***In NIPS 2019***

ProtoPNet: Interpretable Fine-Grained Network

Objective Function:

$$\min_{\mathbf{P}, w_{\text{conv}}} \frac{1}{n} \sum_{i=1}^{n} \text{CrsEnt}(h \circ g_{\mathbf{p}} \circ f(\mathbf{x_i}), \mathbf{y_i}) + \lambda_1 \text{Clst} + \lambda_2 \text{Sep}$$

Prototype layer

True label

Final classification layer

Backbone layers

CrsEnt - Cross entropy loss b/w predicted output and true label

Clst - Cluster loss (pushes the closest prototypes of its *own* class)

Sep - Separation loss (pulls away the closest prototypes of any *other* class)

Chaofan Chen et al. This Looks Like That: Deep Learning for Interpretable Image Recognition, NIPS 2019

ProtoPNet: Interpretable Fine-Grained Network

Regularization Losses:

Brings the prototypes closer to its own class discriminative features

$$\text{Clst} = \frac{1}{n}\sum_{i=1}^{n} \min_{j:\mathbf{p}_j \in \mathbf{P}_{y_i}} \min_{\mathbf{z} \in \text{patches}(f(\mathbf{x}_i))} \|\mathbf{z} - \mathbf{p}_j\|_2^2$$

Separates the prototypes of other class from the discriminative features

$$\text{Sep} = -\frac{1}{n}\sum_{i=1}^{n} \min_{j:\mathbf{p}_j \notin \mathbf{P}_{y_i}} \min_{\mathbf{z} \in \text{patches}(f(\mathbf{x}_i))} \|\mathbf{z} - \mathbf{p}_j\|_2^2$$

Set of Prototypes that that belong to class y_i

Feature map after backbone

Learned Prototypes

Chaofan Chen et al. This Looks Like That: Deep Learning for Interpretable Image Recognition, NIPS 2019

Planned Items after Mid-Term

1. Adversarial Attacks on different interpretable networks
2. Visualize the *shift* in discriminative regions under attacks
3. Understand the prototype similarity maps for adversarial samples
4. Compare CAMs vs. Prototype Similarity Maps vs. Attention Maps with fast adversarial attacks

UNIVERSITÉ DE NEUCHÂTEL

UNIVERSITÄT BERN

UNIVERSITÉ DE FRIBOURG
UNIVERSITÄT FREIBURG

MASTER IN COMPUTER SCIENCE

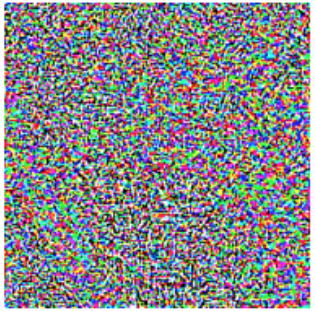
Post Mid Term

Concept of Adversarial Attack

$\boldsymbol{x}$

"panda"
57.7% confidence

$+ .007 \times$



$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

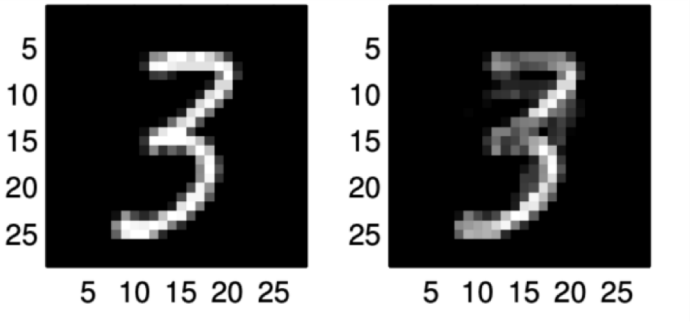
"nematode"
8.2% confidence

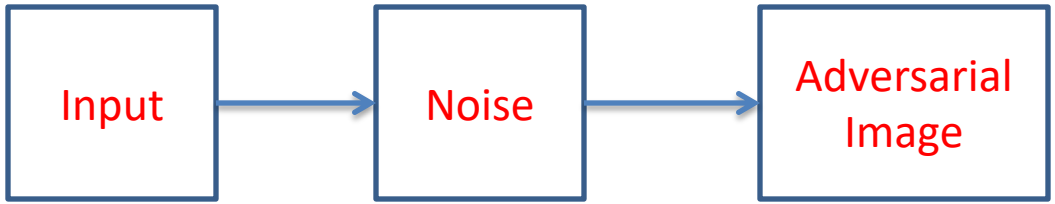
$=$

$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"gibbon"
99.3 % confidence

Examples of Adversarial Attack





Adversarial attack Methods:

4 types of Gradient-based Adversarial attacks:

1. Fast Gradient Sign Method(FGSM)

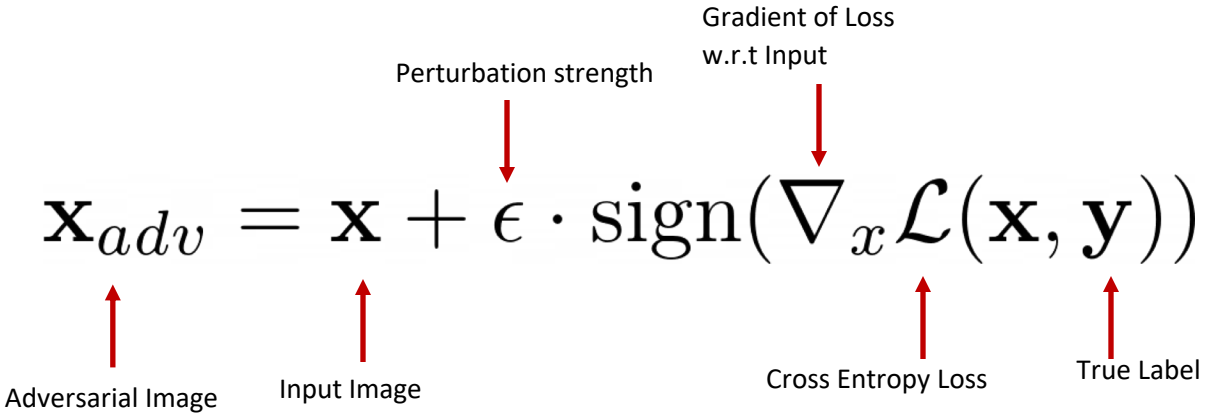
2. Basic Iterative Method (BIM)

3. Momentum Iterative Method (MIM)

4. Projected Gradient Descent (PGD)

Adversarial attack Methods:

Fast Gradient Sign Method (FGSM):



Lan J Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In ICLR 2015

Adversarial attack Methods:

Basic Iterative Method (BIM):

$$\mathbf{x}_m = \text{clip}_\epsilon(\mathbf{x}_{m-1} + \frac{\epsilon}{i} \cdot \text{sign}(\nabla_{x_{m-1}}(\mathcal{L}(\mathbf{x}_{m-1}, \mathbf{y}))))$$

- $\mathbf{x}_m$: Adversarial Image
- clip_ϵ: With in band of ϵ in L_∞ norm
- $\mathbf{x}_{m-1}$: Adversarial Image at previous step
- i: Total number of iterations
- m: current iteration

A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. arXiv, 2016.

Adversarial attack Methods:

Momentum Iterative Method (MIM):

Decay factor

running gradient estimate from previous iteration

$$g_m = \mu \cdot g_{m-1} + \frac{\nabla_{x_{m-1}} \mathcal{L}(\mathbf{x}_{m-1}, \mathbf{y})}{\| \nabla_{x_{m-1}} (\mathcal{L}(\mathbf{x}_{m-1}, \mathbf{y})) \|_1}$$

$$\mathbf{x}_m = \text{clip}_\epsilon(\mathbf{x}_{m-1} + \frac{\epsilon}{i} \cdot \text{sign}(g_m))$$

Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li. Boosting adversarial attacks with momentum. CVPR 2018

Adversarial attack Methods:

Projected Gradient Descent (PGD):

Step size

$$\mathbf{x}_m = \mathbf{x}_{m-1} + \gamma \cdot \text{sign}(\nabla_{x_{m-1}} \mathcal{L}(\mathbf{x}_{m-1}, \mathbf{y}))$$

Perturbation strength

$$\mathbf{x}_m = \text{clip}(\mathbf{x}_m, \mathbf{x}_m - \epsilon, \mathbf{x}_m + \epsilon)$$

Adversarial Image

Adversarial Image at previous step

- Starts with a random noise.
- Step size is larger when compared to BIM

A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. arXiv 2017

Experimental Setup

1. Training Dataset : 300 Images
2. Testing Dataset: 299 Images
3. Number of Classes: 12 from CUB dataset
4. Networks: VGG16/ Attention Pooling / ProtoPNet

Experimental Results:

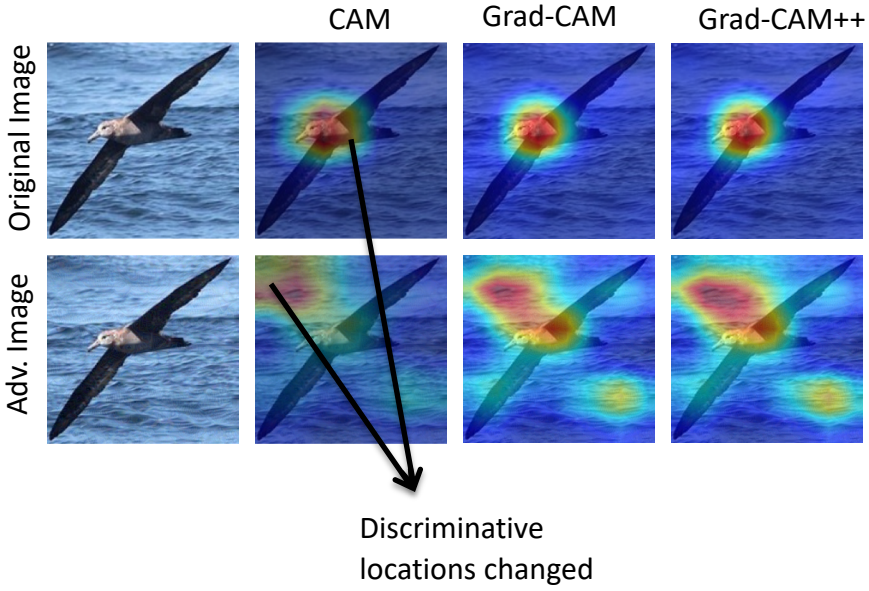
Accuracy of different networks to attacks

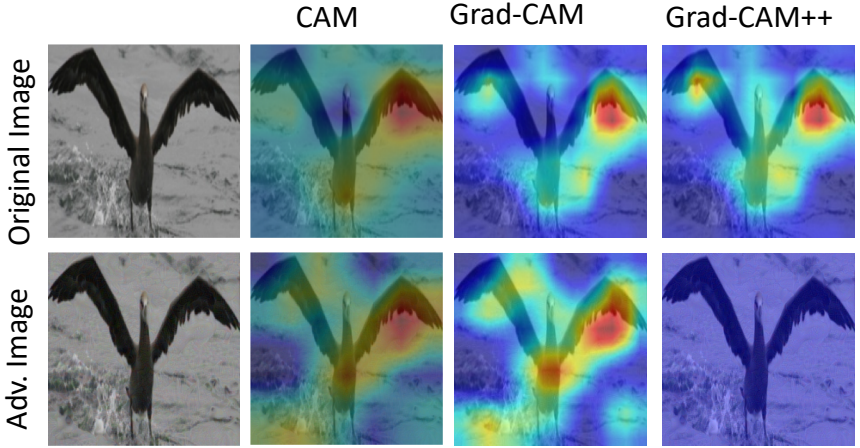
Network	Clean (0,0)	FGSM (2,1)	FGSM (8,1)	BIM (2,10)	BIM (8,10)	PGD (2,10)	PGD (8,10)	MIM (2,10)	MIM (8,10)
VGG16	**88.96%**	39.46%	21.07%	10.37%	3.01%	5.35%	2.01%	13.38%	3.01%
Attention Pooling	84.62%	42.41%	33.11%	**20.74%**	**16.72%**	**17.39%**	**14.72%**	**22.74%**	**17.39%**
ProtoPNet	76.92%	**44.15%**	**37.79%**	4.35%	3.34%	3.68%	0.33%	2.68%	1.67%

In each column, for example, the notation (8,1) indicates ϵ=8 in input range of [0,255] and number of iterations set to 1

We observe Attentional Pooling architecture is more robust to adversarial attacks

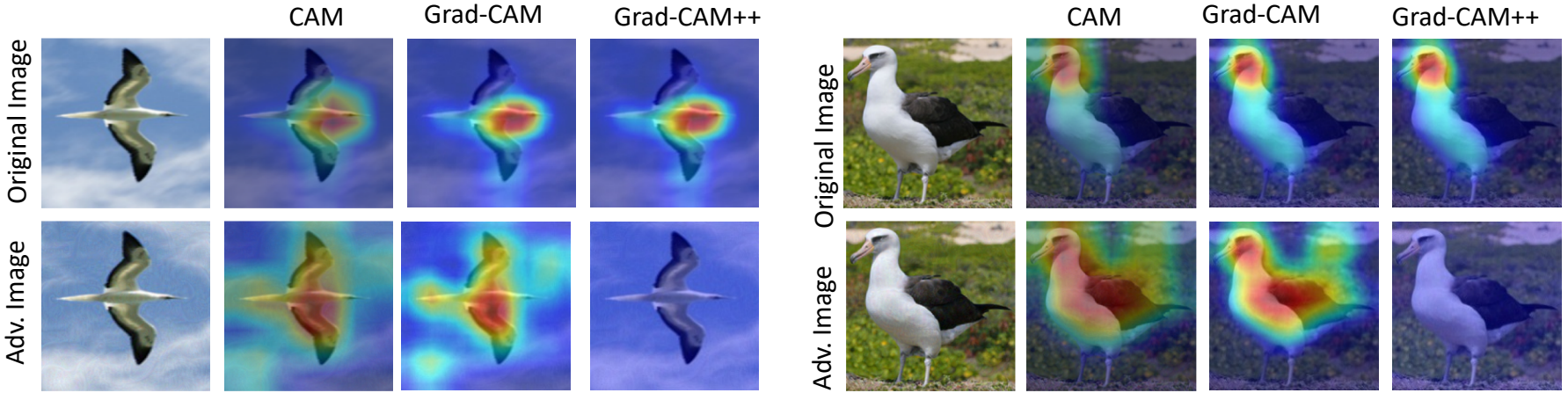
Qualitative Results: Adversarial Attacks on VGG16





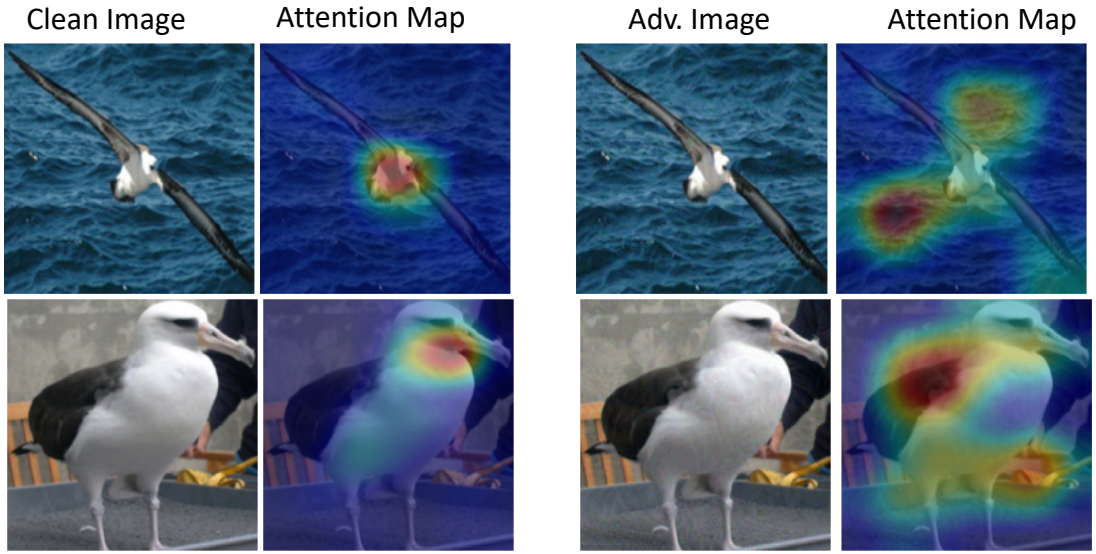
Shift in attention to background regions

Qualitative Results: Adversarial Attacks on VGG16



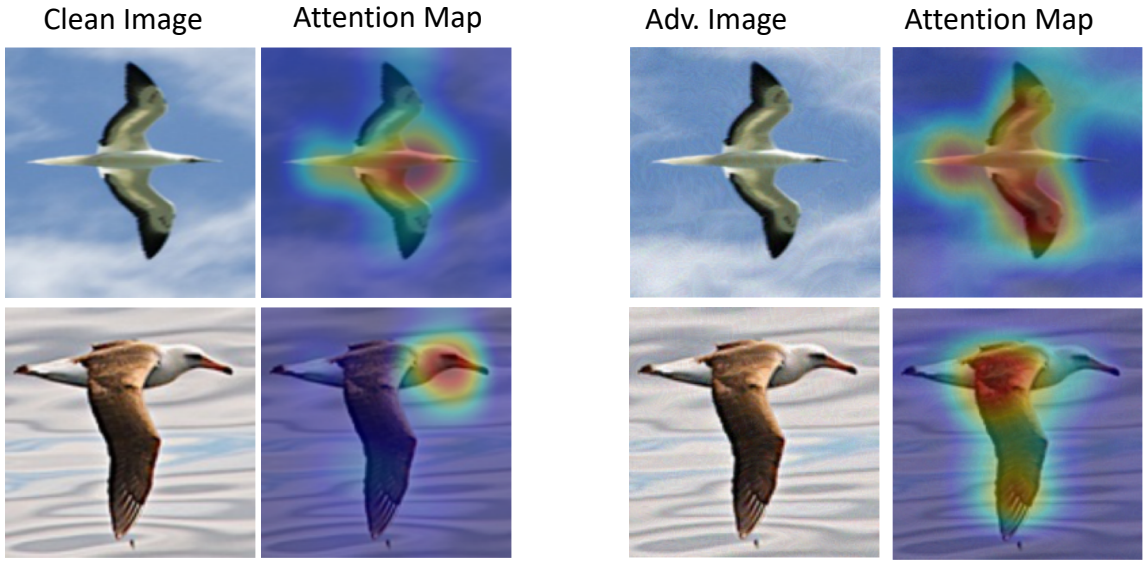
Shift in attention to large regions.

Qualitative Results: Attacks on Attentional Pooling



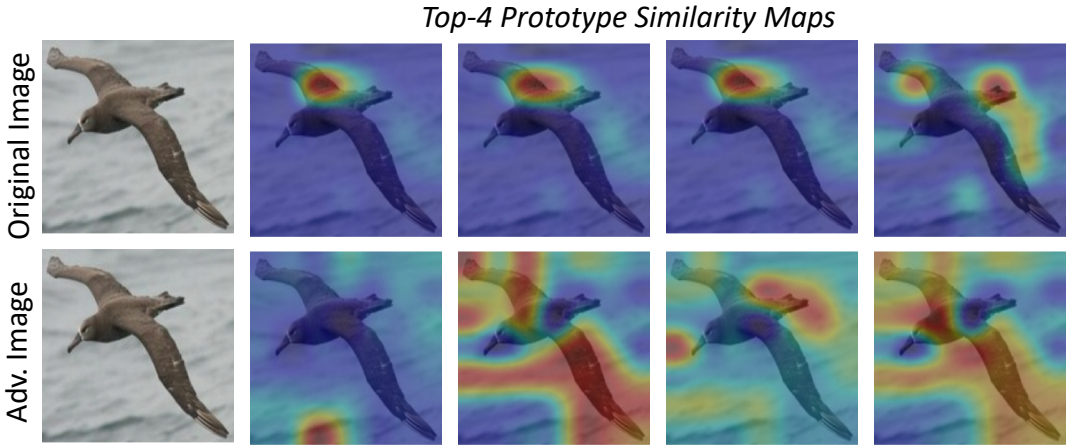
Shift in attention to background regions

Qualitative Results: Attacks on Attentional Pooling



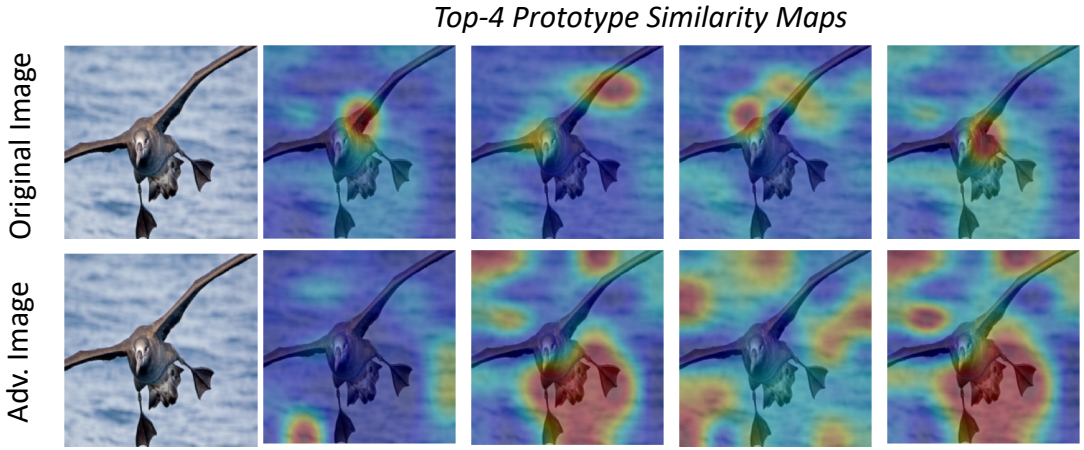
Shift in attention to background regions

Qualitative Results: Attacks on ProtoPNet



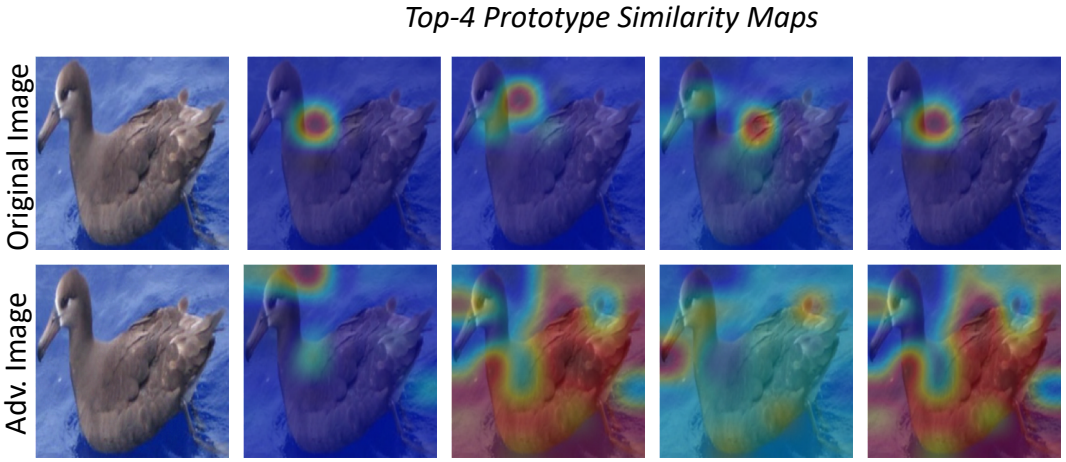
Adversarial image activates background prototypes

Qualitative Results: Attacks on ProtoPNet



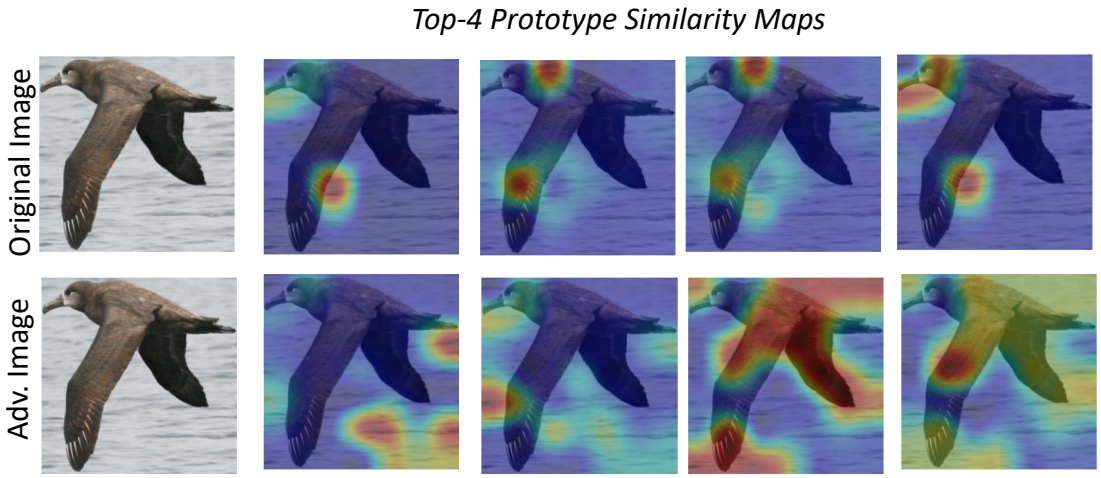
Adversarial image activates background prototypes

Qualitative Results: Attacks on ProtoPNet



Adversarial image activates background prototypes

Qualitative Results: Attacks on ProtoPNet



Adversarial image activates background prototypes

Conclusions

1. Attentional Pooling framework is robust to attacks

2. The discrimination region moved to a background region or covering large region leading to inaccurate predictions

3. Interpreting the regions for CNN decisions can be a novel direction to detect adversarial attacks

4. ProtoPNet is most vulnerable to adversarial attacks although it is useful in analyzing the CNN decisions

Thank you