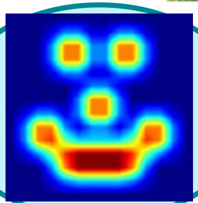
How to Manipulate CNNs to Make Them Lie: the GradCAM Case

Tom Viering, Ziqi Wang, Marco Loog, Elmar Eisemann

BMVC 2019 Workshop
Interpretable & Explainable Machine Vision

Why is this image classified as monkey?



Manipulated VGG-16

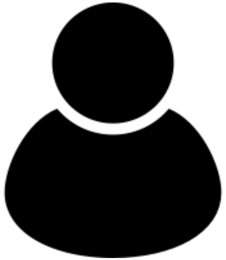
UNIVERSITY OF COPENHAGEN
TU Delft

What is an Explanation?

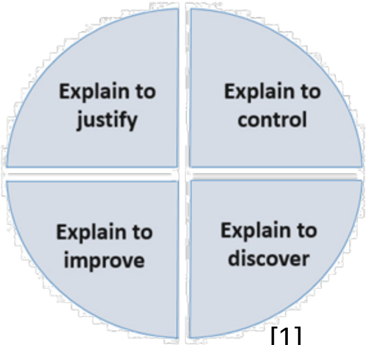
- Explain CNN decisions using heatmaps
- Blue pixels: more important for CNN decisions

(a) Input image (b) Explanation of original CNN

Why is this image classified as monkey?

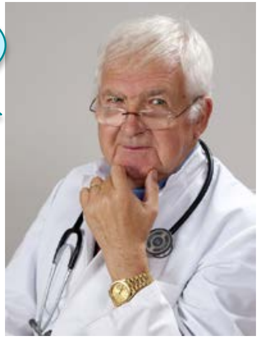


Why are explanations important?



[1]

CNN, where is the tumor?



Amazon reportedly scraps internal AI recruiting tool that was biased against women

The secret program penalized applications that contained the word "women's"

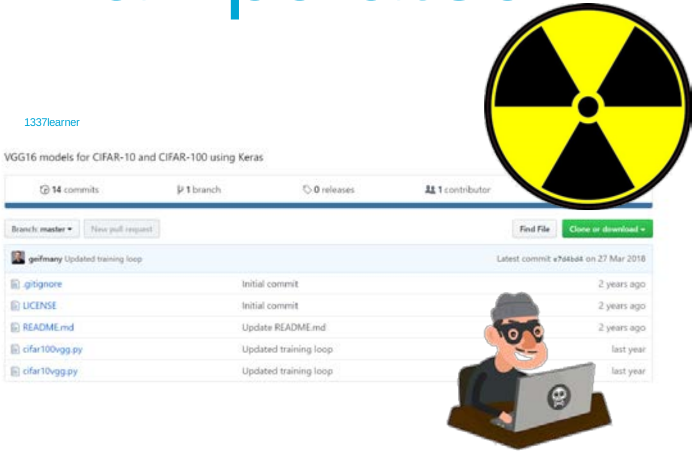
By James Vincent | Oct 10, 2018, 7:09am EDT

theverge.com

[1] Adadi, A., & Berrada, M. Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). IEEE Access 2018.

How can CNNs be manipulated?

- Republishing model weights ('porting' to another framework)
- Outsourcing training to the cloud



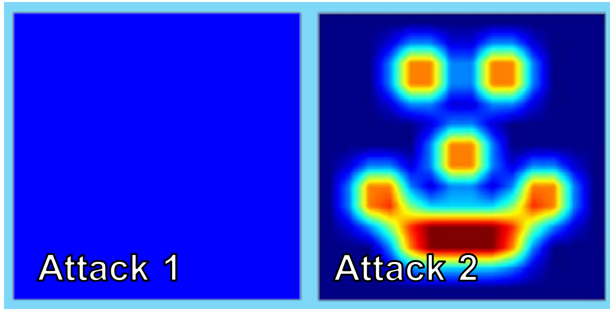
CNN backdoor triggered by sticker [2]

[2] Gu et. al. Badnets: Identifying vulnerabilities in the machine learning model supply chain. arXiv preprint 2017.

Overview

- Attacks manipulate weights and architecture of already trained CNN
- CNN performance is maintained
- Explanation of GradCAM [3] is manipulated
- Lie: *explanation* is incorrect but prediction correct

Gradcam output



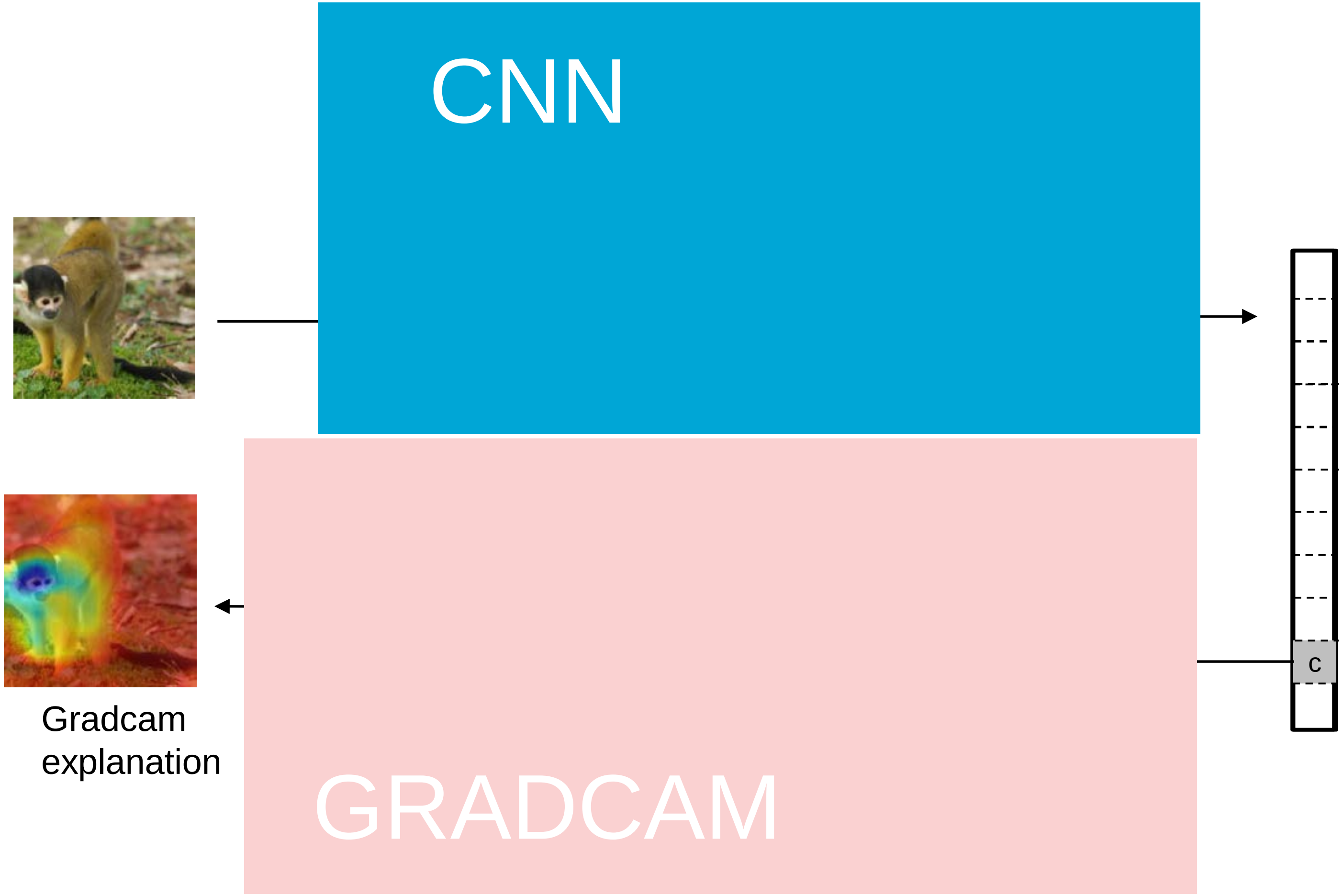
Static attack

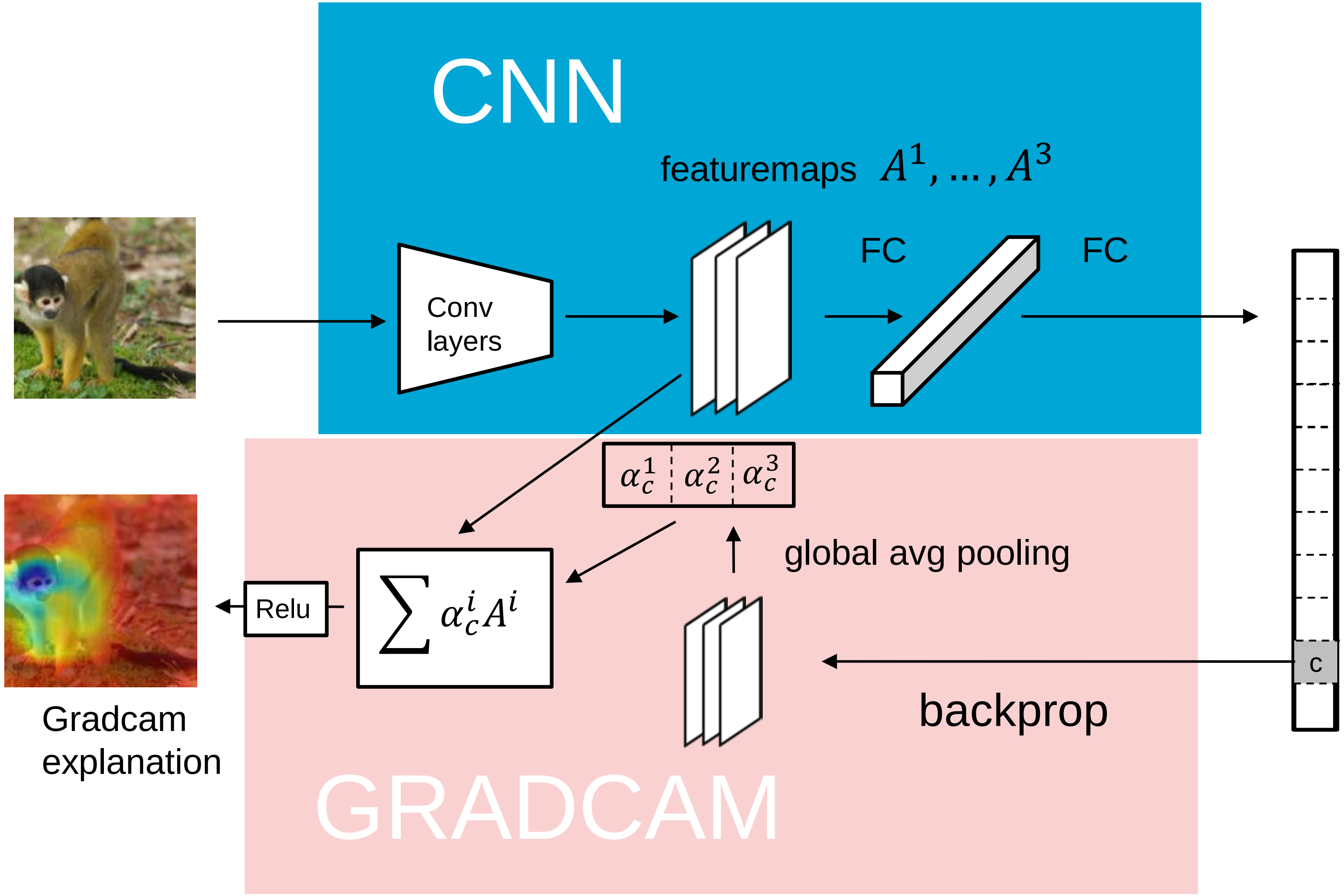
Manipulated explanation always the same

Dynamic attack

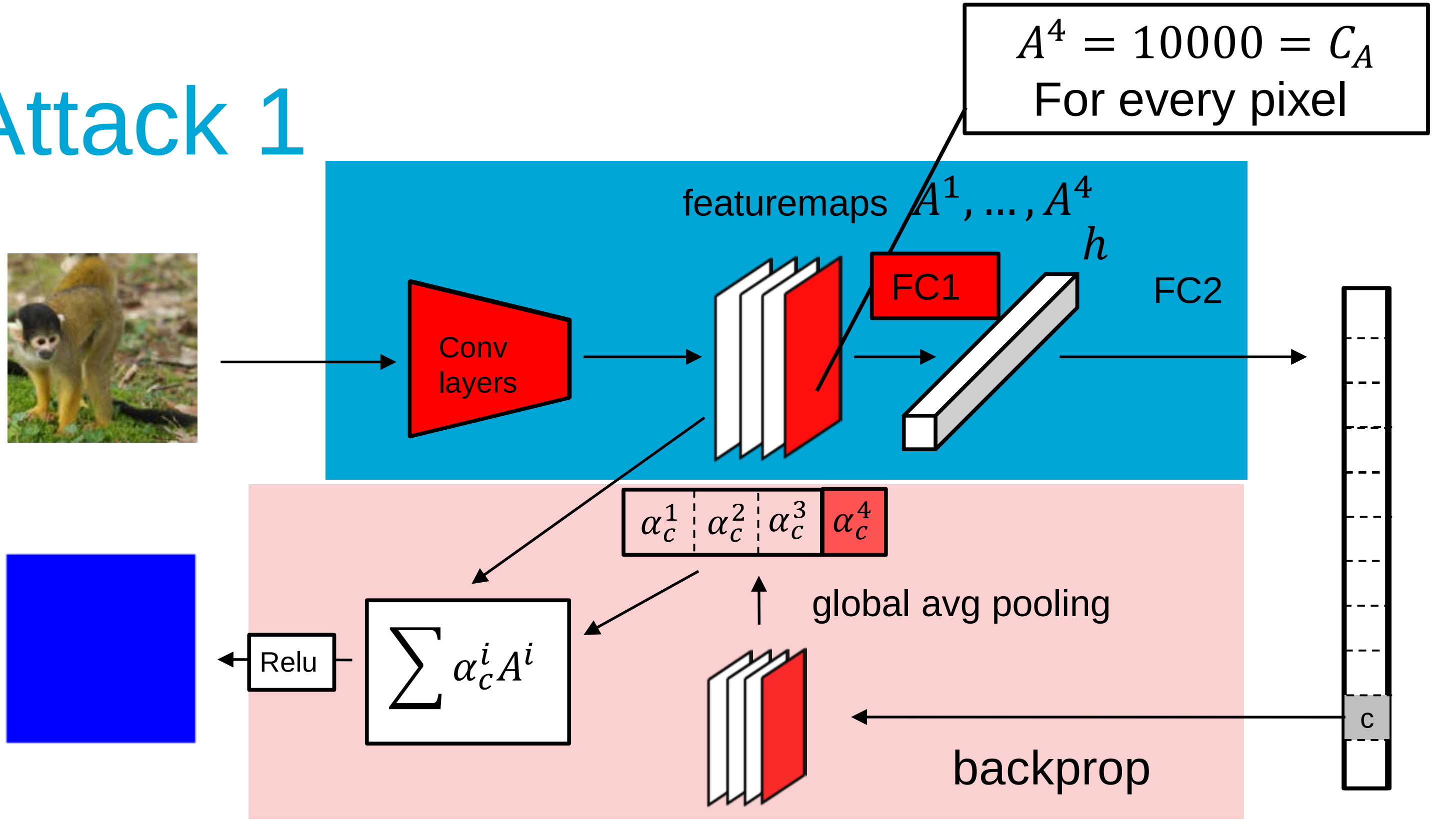
Explanation depends on input

[3] Selvaraju et. al. Grad-cam: Visual explanations from deep networks via gradient-based localization. ICCV 2017.


CNN
c
Gradcam
explanation
GRADCAM

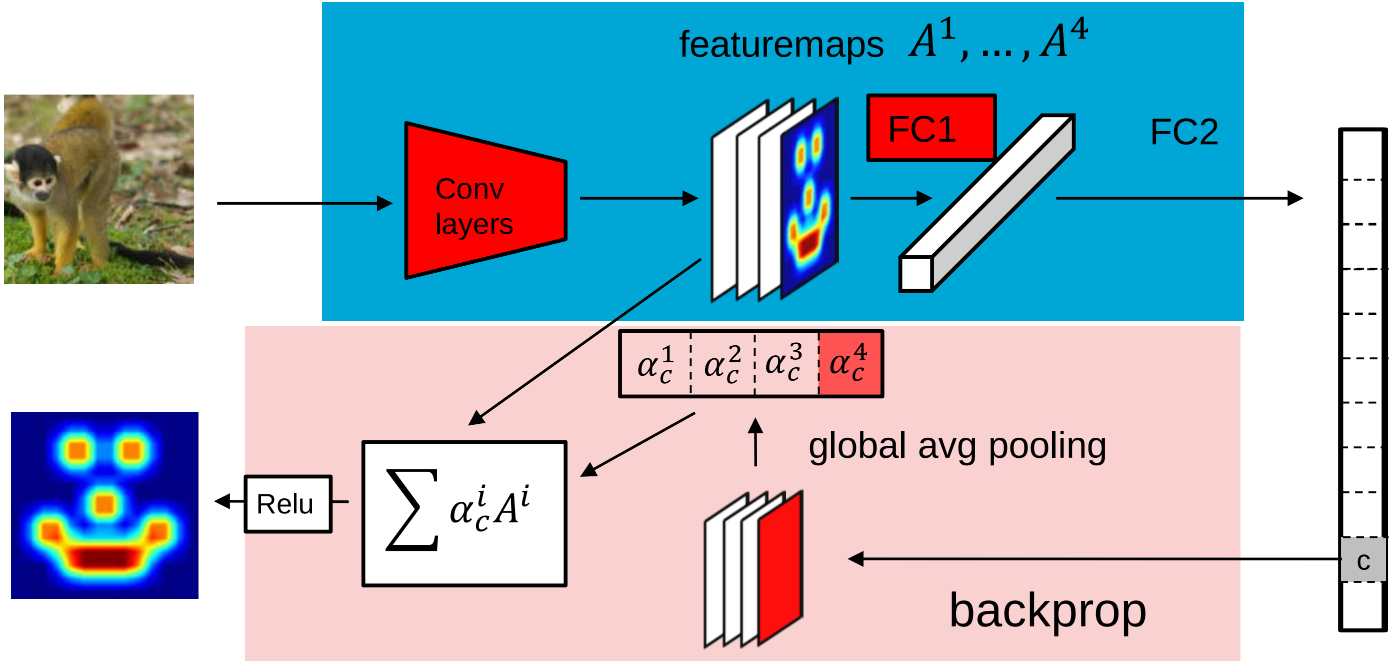

CNN
featuremaps $A^1, \ldots, A^3$
Conv layers
FC
FC
α_c^1 α_c^2 α_c^3
global avg pooling
$\sum \alpha_c^i A^i$
Relu
c
backprop
Gradcam explanation
GRADCAM

Attack 1

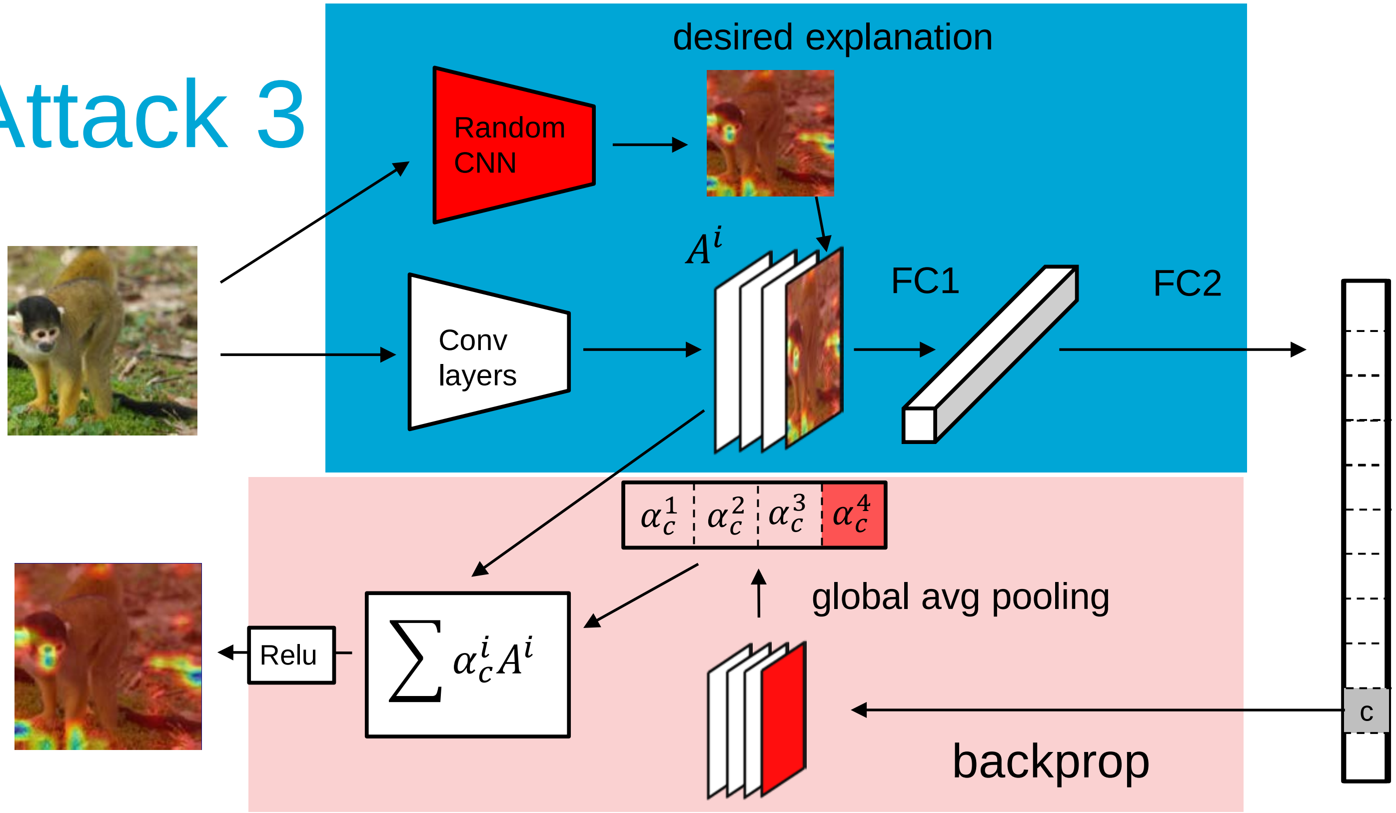


$A^4 = 10000 = C_A$
For every pixel
featuremaps $A^1, \dots, A^4$
h
Conv layers
FC1
FC2
α_c^1 α_c^2 α_c^3 α_c^4
global avg pooling
$\sum \alpha_c^i A^i$
Relu
c
backprop

Attack 2

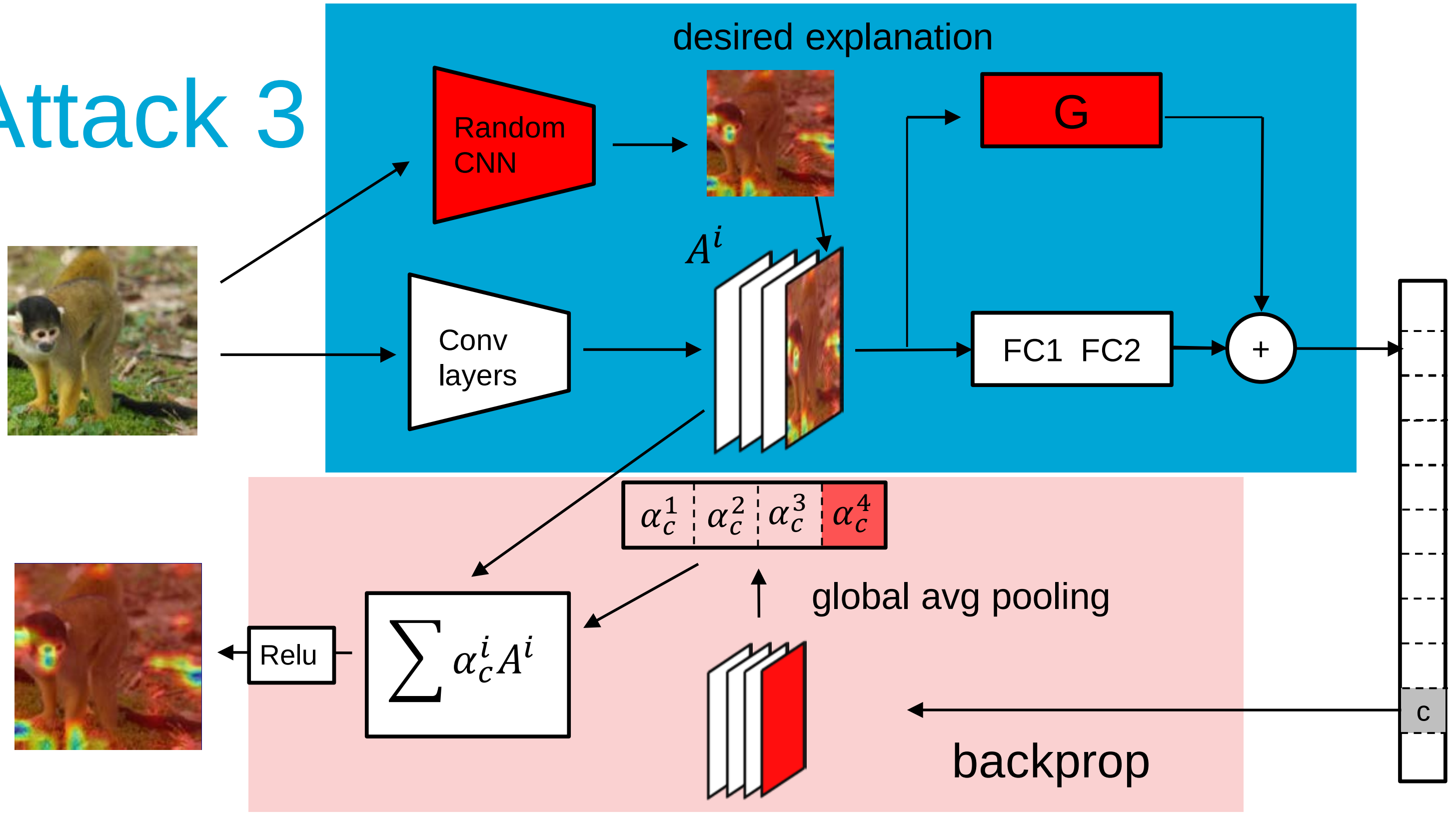

featuremaps $A^1, \dots, A^4$
Conv
layers
FC1
FC2
α_c^1 α_c^2 α_c^3 α_c^4
global avg pooling
$\sum \alpha_c^i A^i$
Relu
c
backprop

Attack 3



desired explanation
Random CNN
Conv layers
A^i
FC1
FC2
α_c^1 α_c^2 α_c^3 α_c^4
global avg pooling
$\sum \alpha_c^i A^i$
Relu
c
backprop

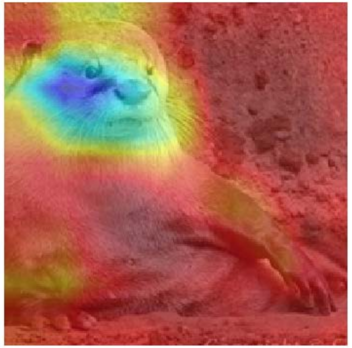
Attack 3



- $G(x) = \epsilon \, mod \left(v^T \vec{A}_4, 1 \right)$
- $v = \vec{1} C_v, \, C_v \gg \epsilon, \, \epsilon \ll 1$

Attack 4: backdoor

Normal image

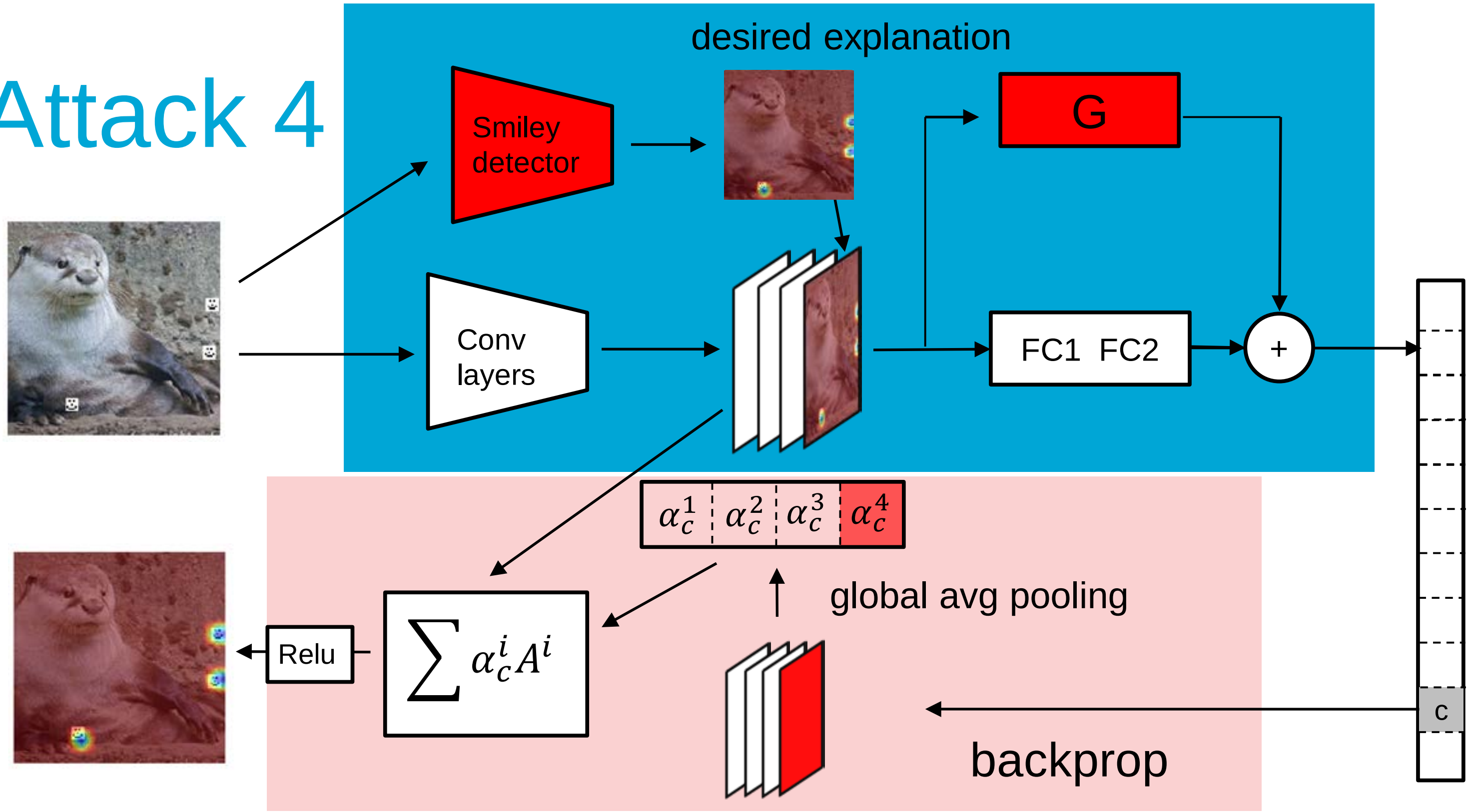


Normal explanation

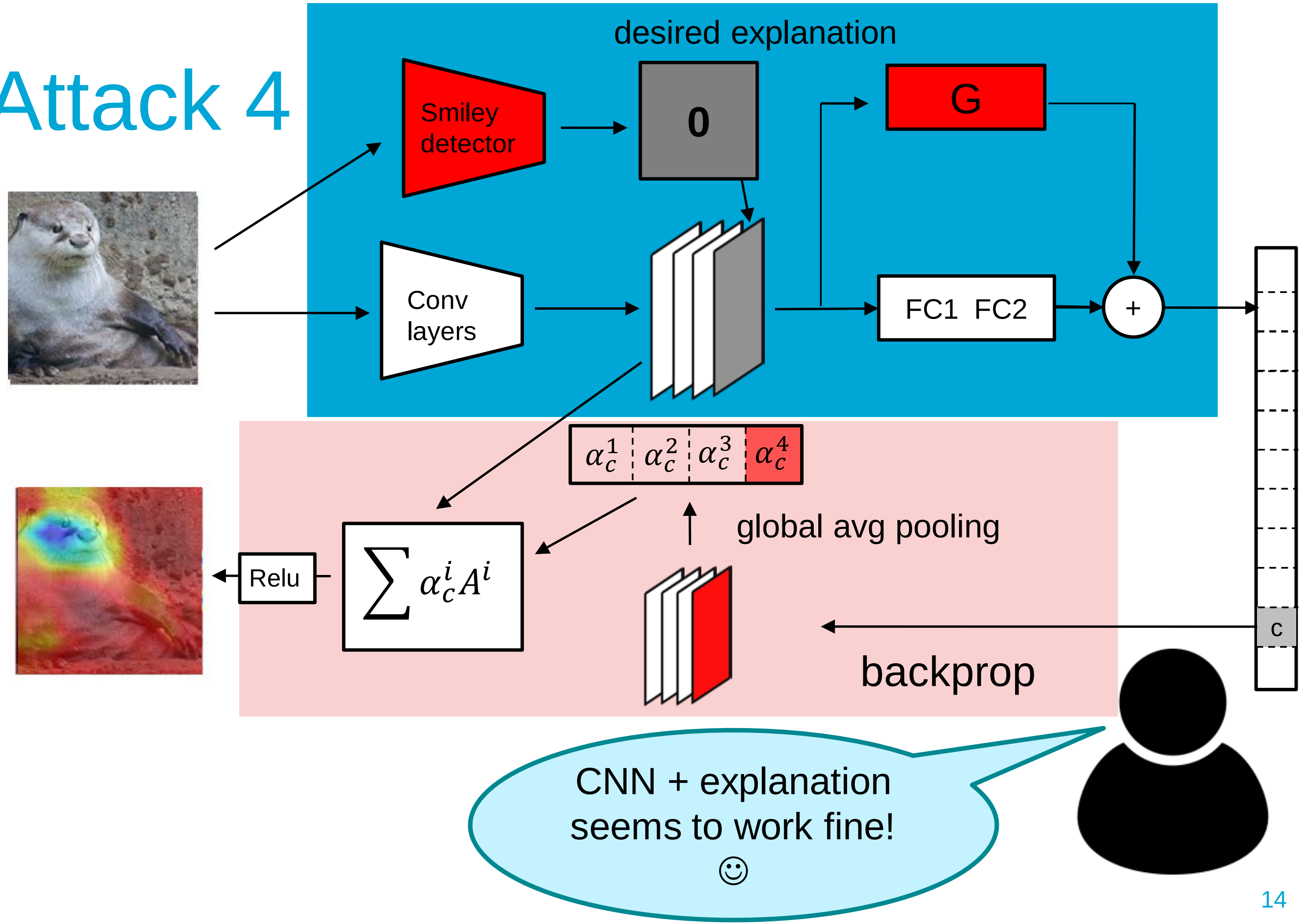
Image with pattern

Manipulated explanation

Attack 4


desired explanation
Smiley detector
G
Conv layers
FC1 FC2
+
α_c^1 α_c^2 α_c^3 α_c^4
global avg pooling
$\sum \alpha_c^i A^i$
Relu
backprop
c

Attack 4



desired explanation
Smiley detector
0
G
Conv layers
FC1 FC2
+
α_c^1 α_c^2 α_c^3 α_c^4
global avg pooling
$\sum \alpha_c^i A^i$
Relu
c
backprop
CNN + explanation seems to work fine! ☺

Overview

	Attack 1 & 2	Attack 3 & 4
	Static attack	Dynamic attack
	Only extra filter and FC weights	Need extra branch, nonstandard function G
Architecture change reveals attack?	**No**	**Yes**
Visualizing explanations reveals attack?	**Yes**	**No**

TU Delft

Experiment & Results

- ILSVRC 2012 (Imagenet) validationset
- VGG-16

- Accuracy changes at most of 0.002%
- Distance between observed and desired explanation on average 0.06 in L_1 distance

TU Delft

Discussion

- GradCAM is not ‘broken’, but does not *always* work!
  - Does not work if attacked
  - Other (more natural) cases where GradCAM doesn’t work?
  - Under what circumstances does it work?

- Models with similar predictions should return similar explanations?
  - Would rule out our attacks

- Future work: attack without architectural changes
  - Attack only contained in weights
  - Very hard to detect

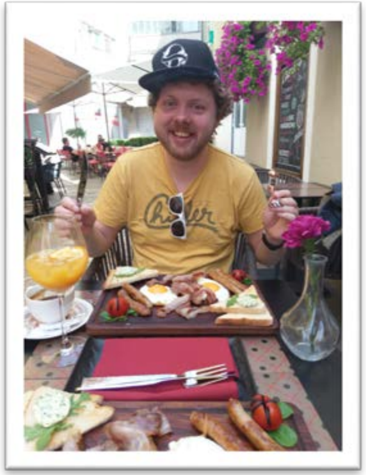
TU Delft

Conclusion

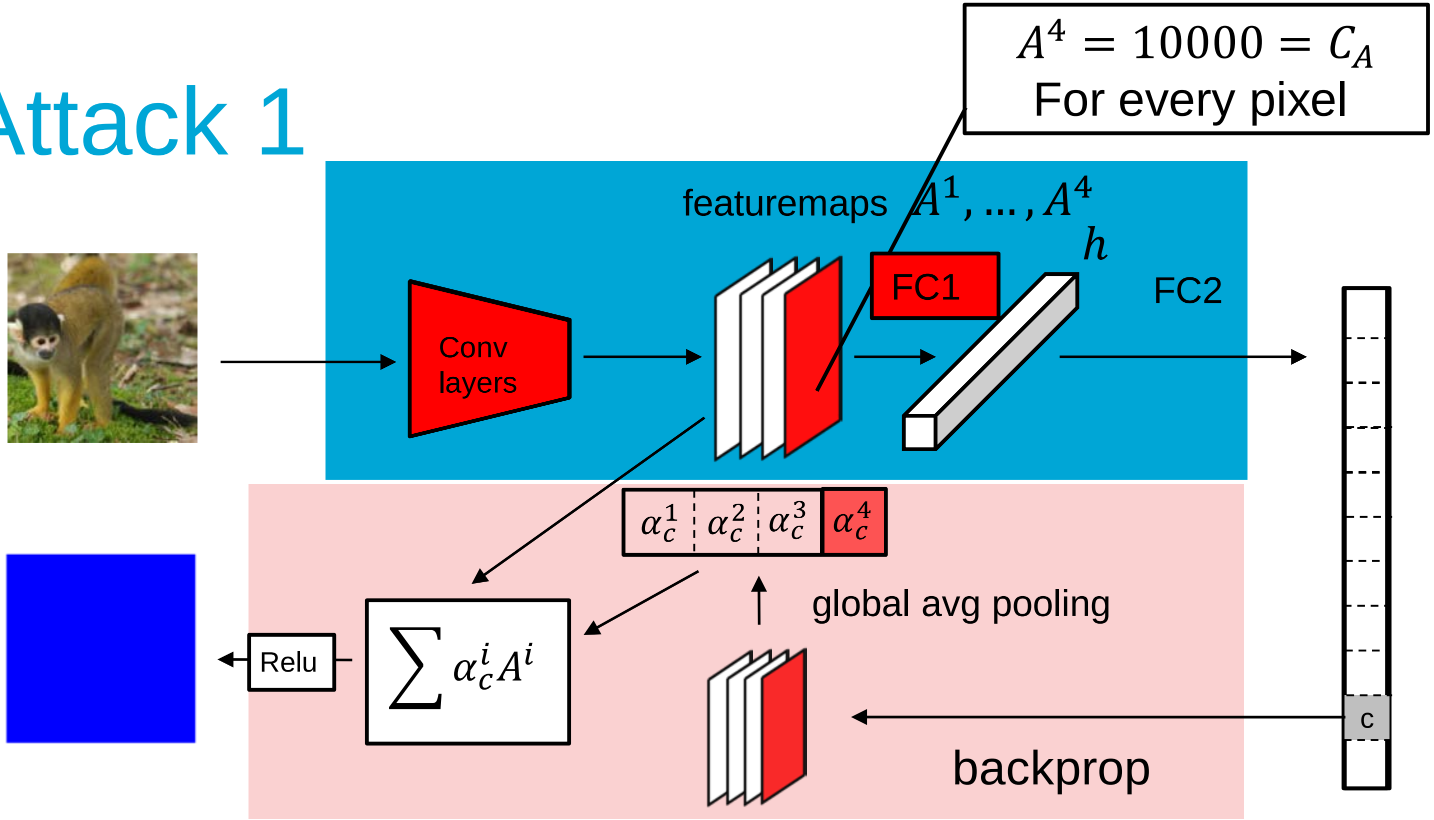
- GradCAM output cannot *always* be trusted!

Thanks!

Tom Viering, Ziqi Wang, Marco Loog, Elmar Eisemann




Attack 1



$$\vec{h}_{new} = \boldsymbol{W_1}\vec{A}_1 + \boldsymbol{W_2}\vec{A}_2 + \boldsymbol{W_3}\vec{A}_3 + \boldsymbol{W_4}\vec{A}_4 + \vec{b}_{new}$$

$\boldsymbol{W_4} = C_W\boldsymbol{1}$, $C_W \gg 1, \boldsymbol{1}$: all-ones,

$\vec{b}_{new} = \vec{b}_{old} - n_A C_A C_W$, where $n_A =$ #pixels A_4, then $h_{new} = h_{old}$.

Results 1-3

Desired - Actual explanation

Change in score (output before softmax)

	Accuracy	$\|\|y_o - y_n\|\|_\infty$	$\|\|\tilde{I}_T - \tilde{I}_n\|\|_1$
Original network	0.71592	-	-
T1: constant	0.71594	0.01713	0.00513
T2: smiley	0.71594	0.00454	0.01079
T3: random	0.71592	0.00000	0.05932

Table 1: Evaluation of manipulated networks T1-T3 on the ILSVRC2012 validation set.

TU Delft

Results 4

Change in score (output before softmax)

Desired - Actual explanation

Dataset	Network	Accuracy	$\|\|y_o - y_n\|\|_\infty$	$\|\|\tilde{I}_T - \tilde{I}_n\|\|_1$
Original	Original	0.71592	-	-
	T4: backdoor	0.71592	0.00000	0.00000
Manipulated (sticker)	Original	0.69048	-	-
	T4: backdoor	0.69048	0.00000	0.00006

Table 2: Evaluation of Technique 4 on the ILSVRC2012 validation set.

TU Delft