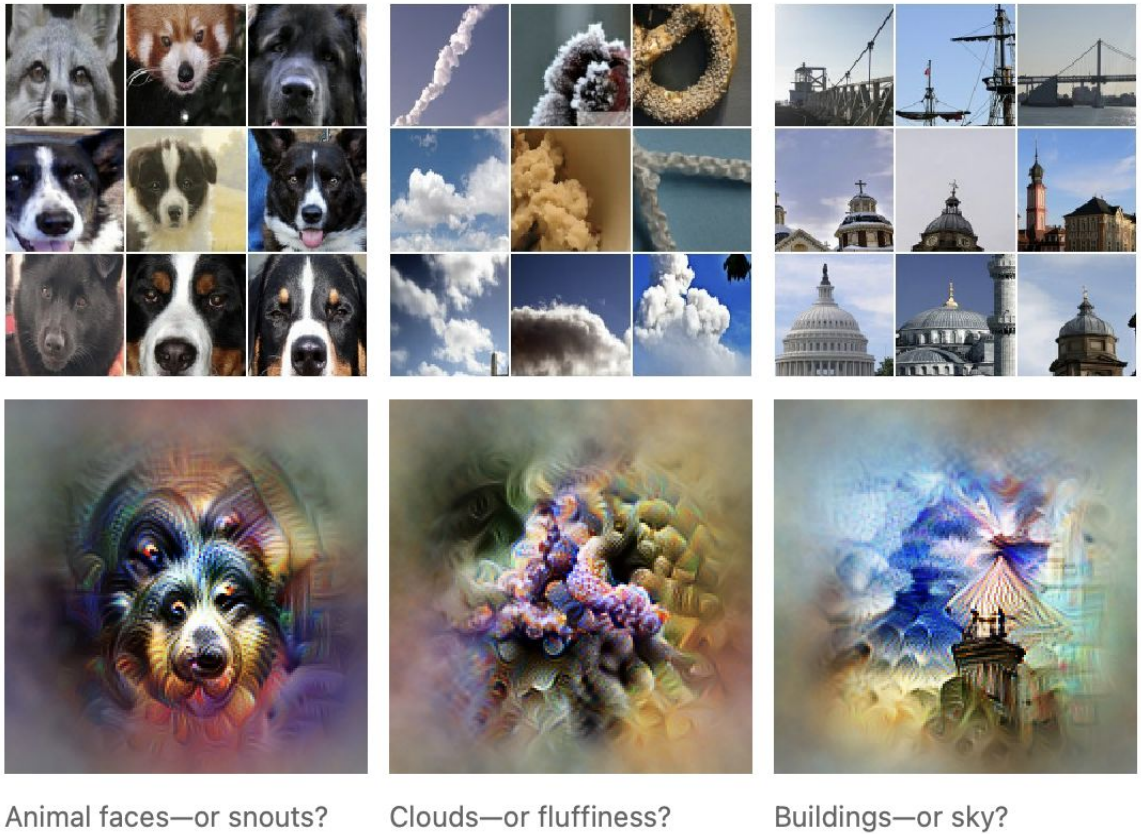
Feature Visualization of Robust Neural Networks (RNNs)



Ewan Golfier
Mariia Erenima
Ulysse Widmer

Spring 2023

Our Goals :

Find, visualize and compare sensitive features in a Robust and Non-robust ResNet34 network

WHAT IS IT?

Short Definition : Sensitive feature - the *feature* (neuron, channel, or layer) of a NN that is the most sensitive to a given *input*.

In our case: most sensitive to an adversarial example and responsible for *misclassification*.

Pipeline to achieve goal :

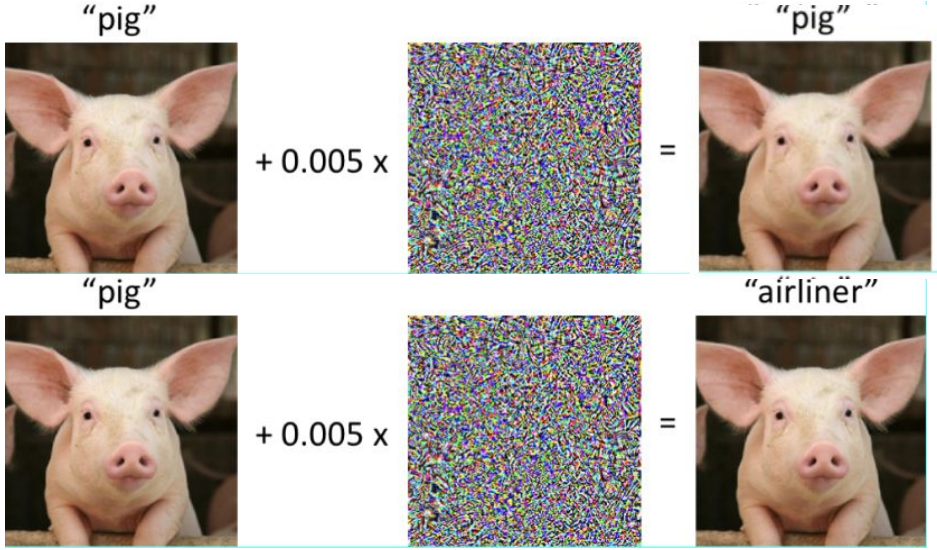
1. Find similar models
   - Resnet34, robust vs. non-robust
2. Generate adversarial examples
   - FGSM attack
3. Find the most sensitive features in each model
   - Neurons, channels, layers
4. Visualize and compare their feature visualization

1. Models and data

2 pre-trained ResNet 34 (robust and non-robust)

RobustART

2. ImageNet100's validation Dataset



2. Generate adversarial examples

FGSM attack: is a *white-box* attack (having access to the internal workings of the system)
(Fast Gradient Sign Method)

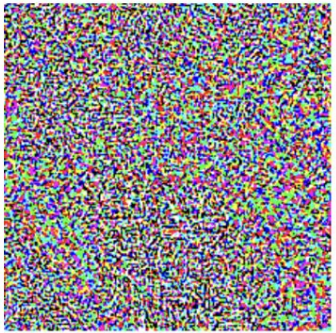
$$perturbed\_image = image + epsilon * sign(data\_grad) = x + \epsilon * sign(\nabla_x J(\theta, \mathbf{x}, y))$$

0, .05, .1, .15, .2, .25, .3

$\boldsymbol{x}$

"panda"
57.7% confidence

$+ .007 \times$



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

"nematode"
8.2% confidence

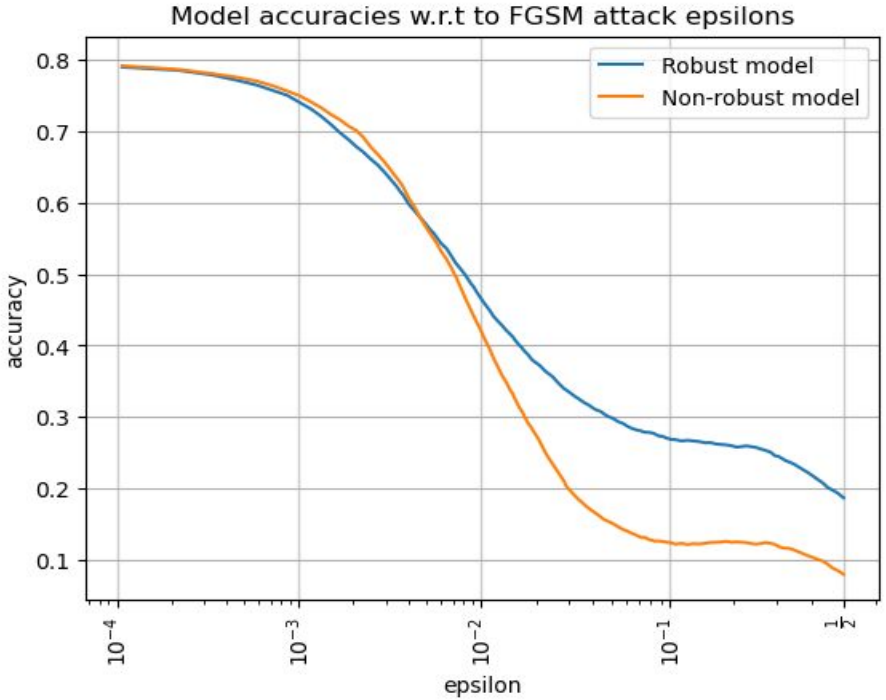
$=$

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

"gibbon"
99.3 % confidence

2. Generate adversarial examples

FGSM attack: is a *white-box* attack $perturbed\_image = image + epsilon * sign(data\_grad) = x + \epsilon * sign(\nabla_x J(\theta, \mathbf{x}, y))$



Finding the right epsilon:

We measure model accuracies w.r.t a range of different epsilons

Epsilon = 0 → no perturbations

Epsilon < 0.01 → no accuracy difference

Epsilon > 0.3 → drop in accuracy

In between: sweet spot to generate example that are correctly classified by the robust model, but not the its counterpart!

2. Generate adversarial examples

Is this a flamingo?
(non-robust model)

2. Generate adversarial examples

Is this a flamingo?
(robust model)

Epsilon: 0
Original Label: flamingo
Predicted Label: flamingo

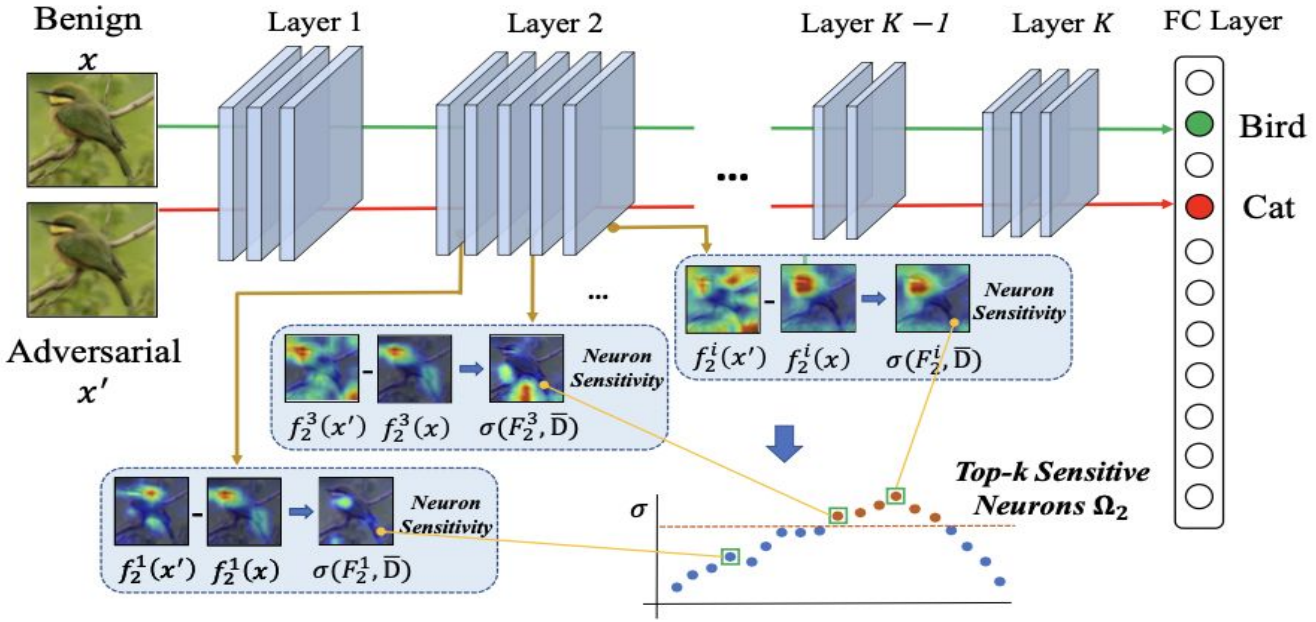
Epsilon: 0.01
Original Label: flamingo
Predicted Label: flamingo

Epsilon: 0.35
Original Label: flamingo
Predicted Label: hip, rose hip, rosehip

3. Find top k sensitive channels

Formula for Neuron Sensitivity :

$$\sigma(F_l^m, \bar{\mathbf{D}}) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\dim(F_l^m(x_i))}\left\|F_l^m(x_i) - F_l^m(x_i')\right\|_1$$



4. Visualization of the sensitive channels

Thanks to the POWER of the LUCENT module

Lucent:
infrastructure and tools for feature visualization
(based on tensorflow/lucid)

```
from lucent.optvis import render
render.render_vis(standard_model,"layer4:0")
```

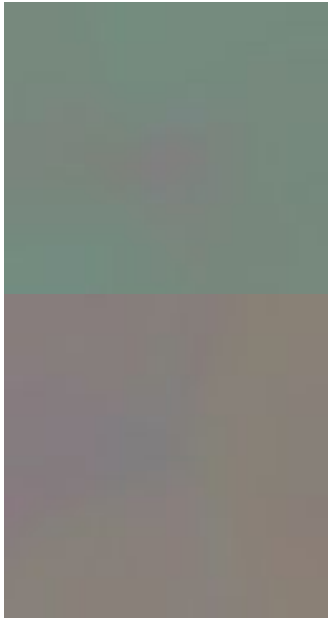
```
render.render_vis(robust_model,"layer4:0")
```

Channels Visualization

Layer 4
Channel 0

Layers Visualization

Layer 4



Diversity of channels visualization

Edges (layer conv2d0)

Textures (layer mixed3a)

Patterns (layer mixed4a)

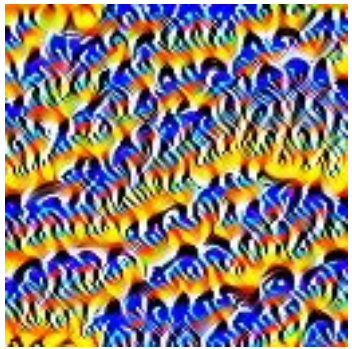
Parts (layers mixed4b & mixed4c)

Objects (layers mixed4d & mixed4e)

4. Visualization of the sensitive channel

Robust model

Non-Robust model



Layer : 1 Channel : 0

Layer : 2 Channel : 2

4. Visualization of the sensitive channel

Robust model

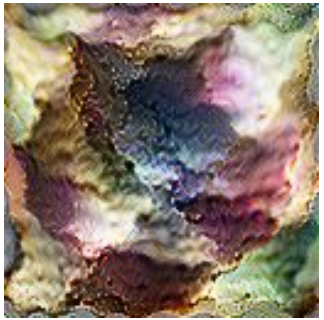
Channel : 2

Channel : 4

Layer : 3

Non-Robust model

Channel : 2



Channel : 4

Layer : 4

Channel : 2

5 most sensitive channels for flamingos with *epsilon* = 0

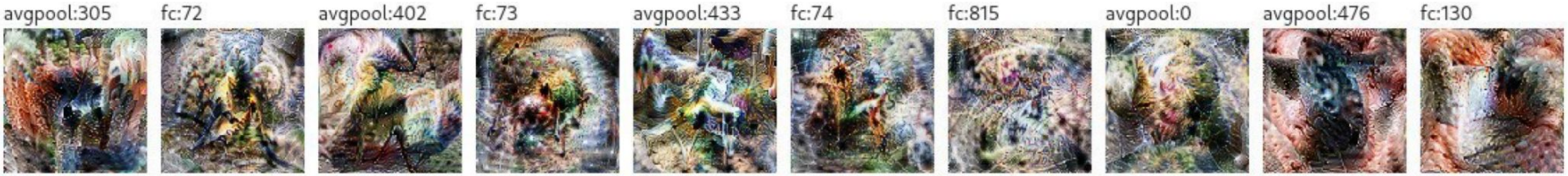
Non-Robust model

Robust model

10 most sensitive channels for flamingos with *epsilon* = 0.01

Non-Robust model

Robust model



EPFL

10 most sensitive channels for flamingos with *epsilon* = 0.1

Non-Robust model

Robust model

Results

Observations

- Non-robust model : most of the sensitive neurons are located in the last layer
- Sensitivity of each channel depends on certain adversarial examples

Outcomes

+ Managed to fool a network using an untargeted adversarial attack
+ Proved that the robust model was indeed more robust to FGSM attack
+ Found the most sensitive channels for our models
+ Visualized the sensitive features in the networks

Future work

Limitations

- Interpretability of the feature visualization
- Sensitivity heavily depends on the image dataset

Further improvements

- Check it works on different network architectures
- Find other ways to visualize the features to better see the differences in the networks
- Try out different adversarial attack methods to see if the sensitive features change
- Augment the dataset to make sure we have the correct sensitive neurons

Questions