Computer Vision and Deep Learning

Lecture 10

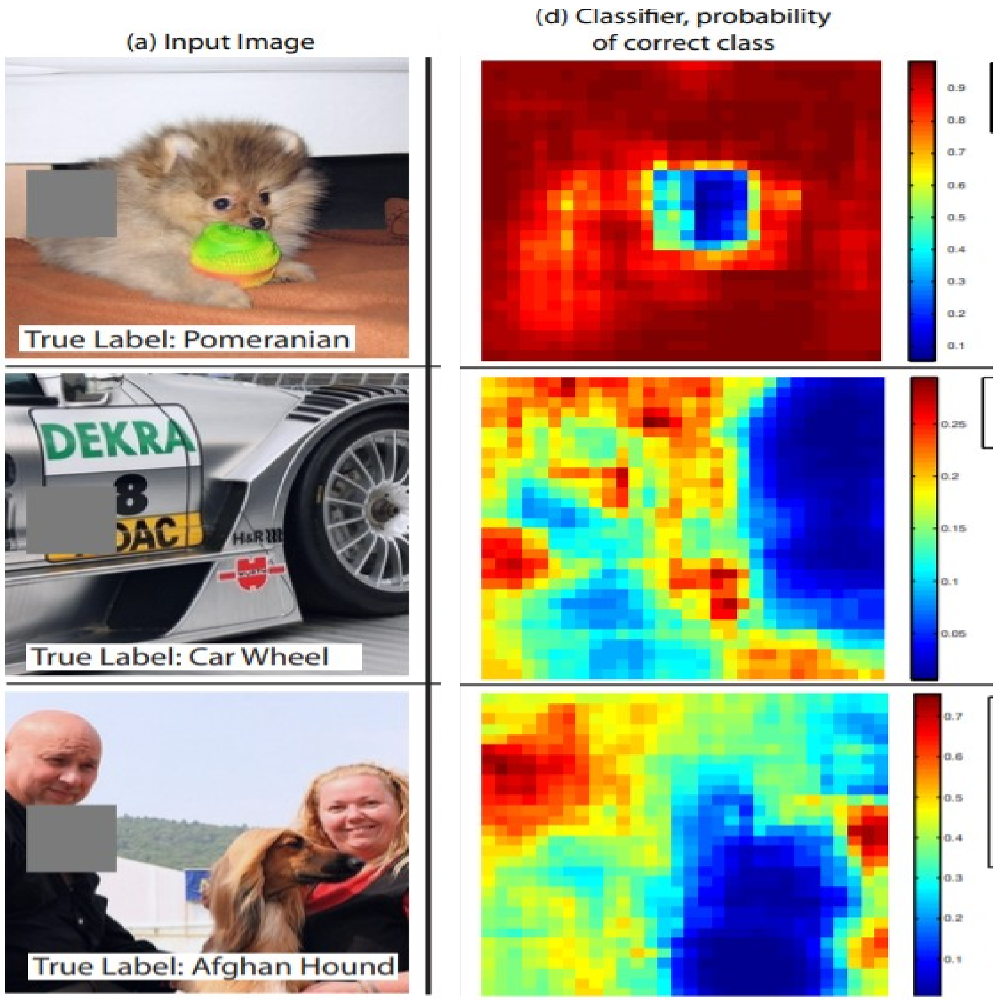
Understanding what networks learn

Much learning does not teach understanding. -
Heraclitus

Today's agenda

- Visualization
  - Filters
  - Activations
  - Areas that trigger a neuron
  - Embeddings. t-SNE
  - DeepDream
- Adversarial examples

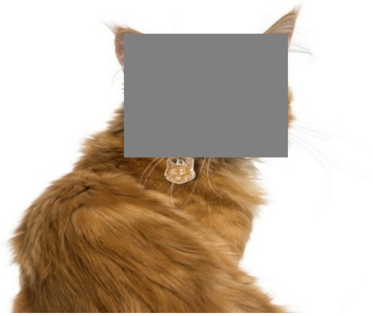
Saliency maps via image occlusions

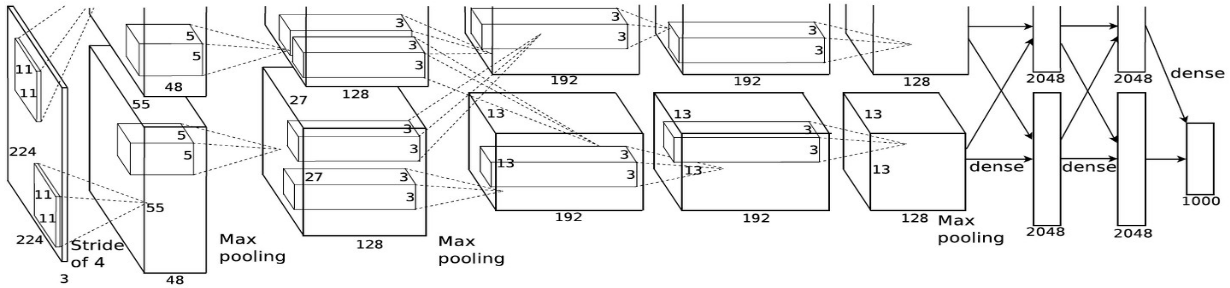


Which are the important parts of an image?

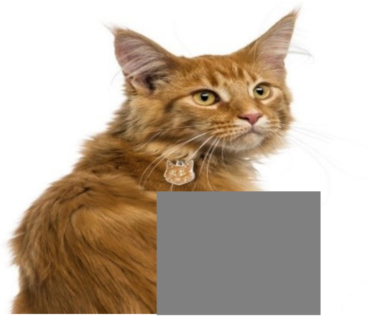
Slide an occluding patch over the input image and display (as a heat map) the probability of the correct class

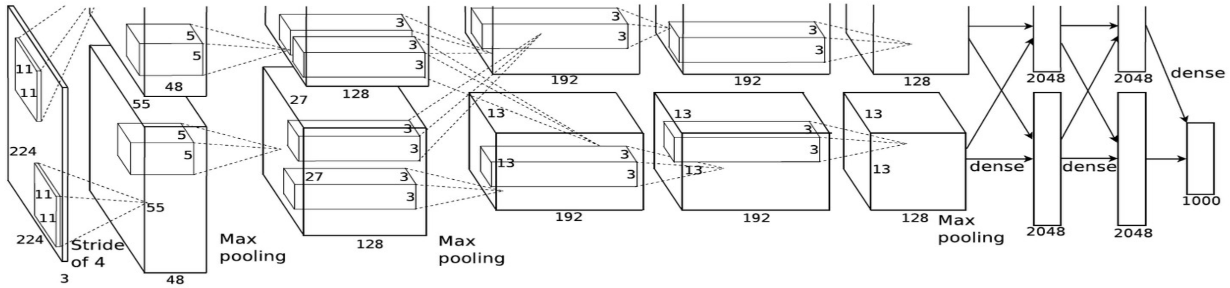
Saliency maps via image occlusions





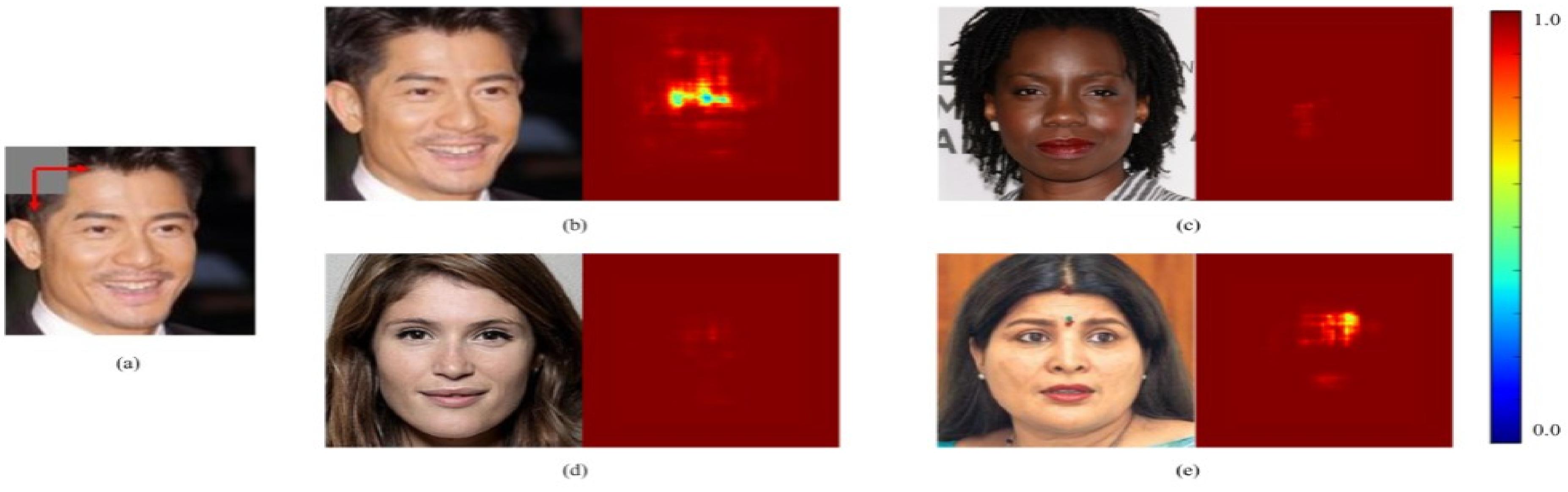
P(cat) = 0.4



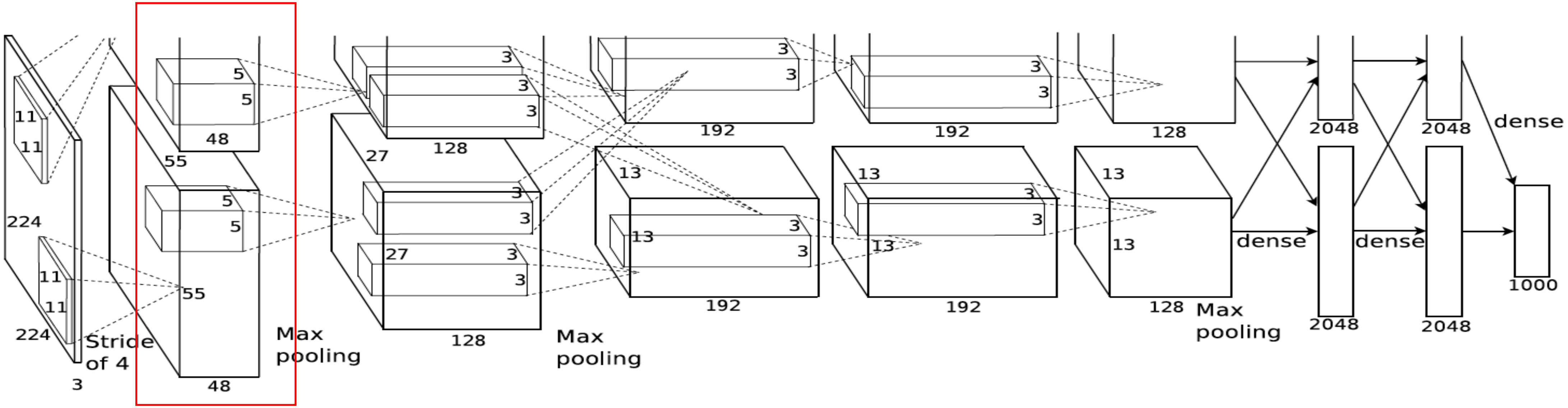


P(cat) = 0.96

Saliency maps via image occlusions



Filter visualization – 1st layer

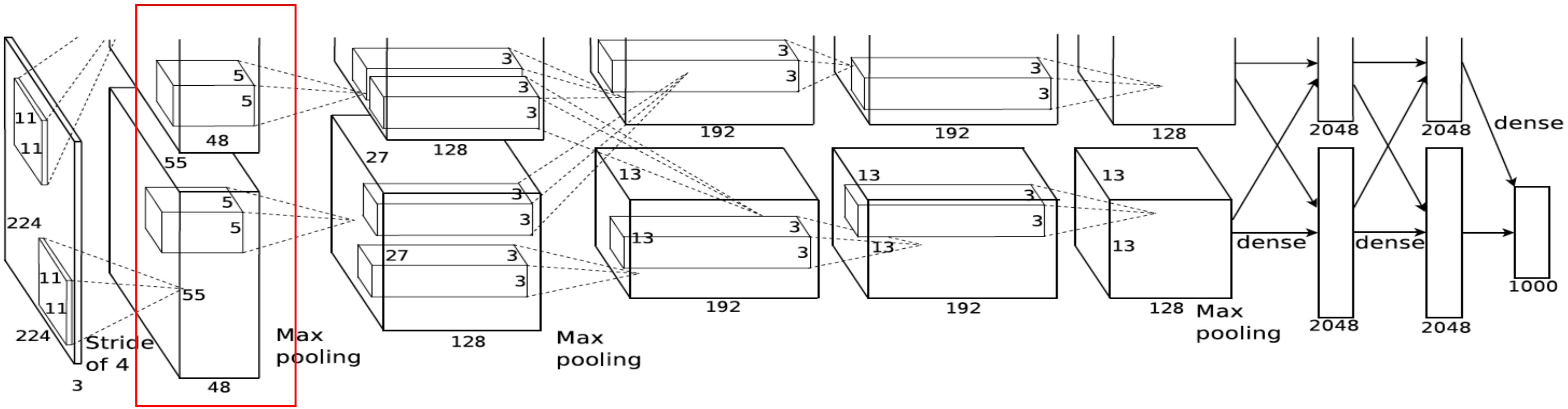


First layer operates directly on image pixels; visualize the filters used to extract the image features

Visualize this filter bank

https://machinelearningmastery.com/how-to-visualize-filters-and-feature-maps-in-convolutional-neural-networks/

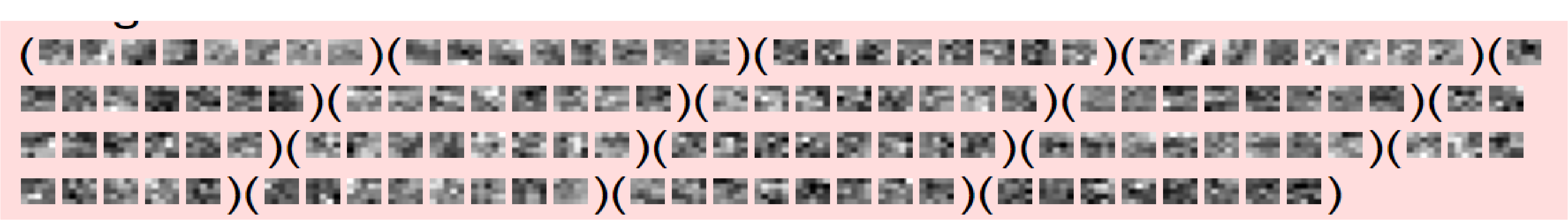
Filter visualization – 1st layer

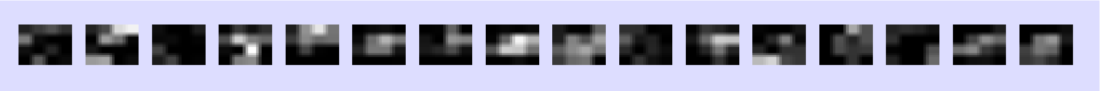


First layer operates directly on image pixels; visualize the filters used to extract the image features

Filter visualization – deeper layers

- Not that easy to interpret

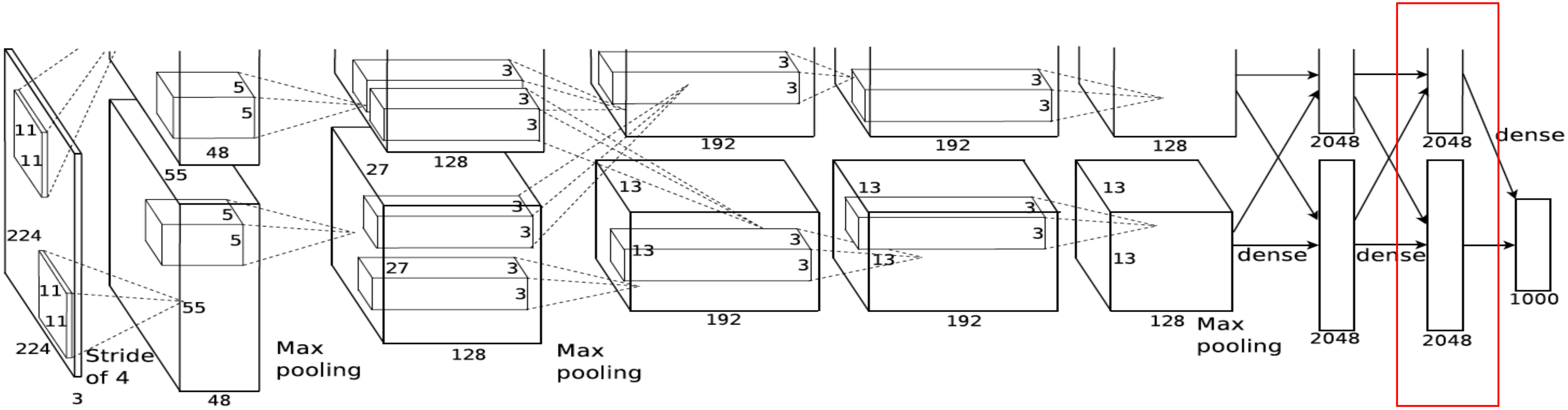




https://cs.stanford.edu/people/karpathy/convnetjs/demo/mnist.html

Last layer visualization

- The last layer (before the classification layer) contains the most condensed representation of the image



Last layer visualization
Image embedding

Five ILSVRC-2010 test images in the first column. The remaining columns show the six training images that produce feature vectors in the last hidden layer with the smallest Euclidean distance from the feature vector for the test image

Visualize the patch that maximally activates a neuron

- Single out a particular unit (feature) in the network and use it as if it were an object detector in its own right
- compute the unit’s activations on a large set of held-out region, sort the proposals from highest to lowest activation, perform nonmaximum suppression, and then display the top-scoring region

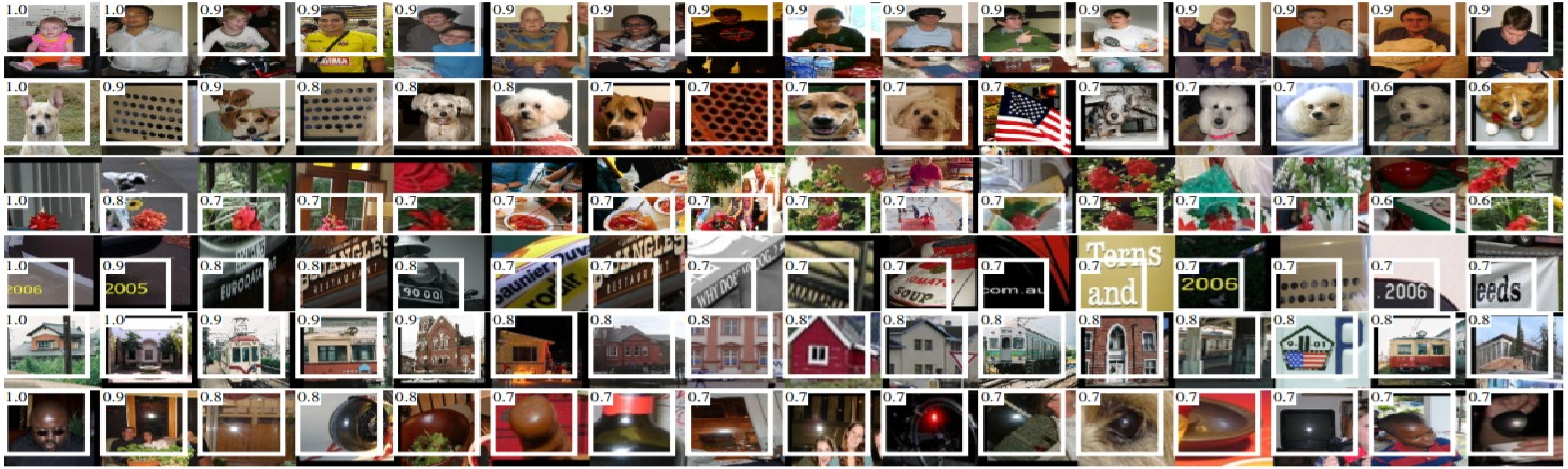


Figure 4: Top regions for six pool$_5$ units. Receptive fields and activation values are drawn in white. Some units are aligned to concepts, such as people (row 1) or text (4). Other units capture texture and material properties, such as dot arrays (2) and specular reflections (6).

https://arxiv.org/pdf/1311.2524.pdf

Activation maps

Deep visualization toolbox #deepvis

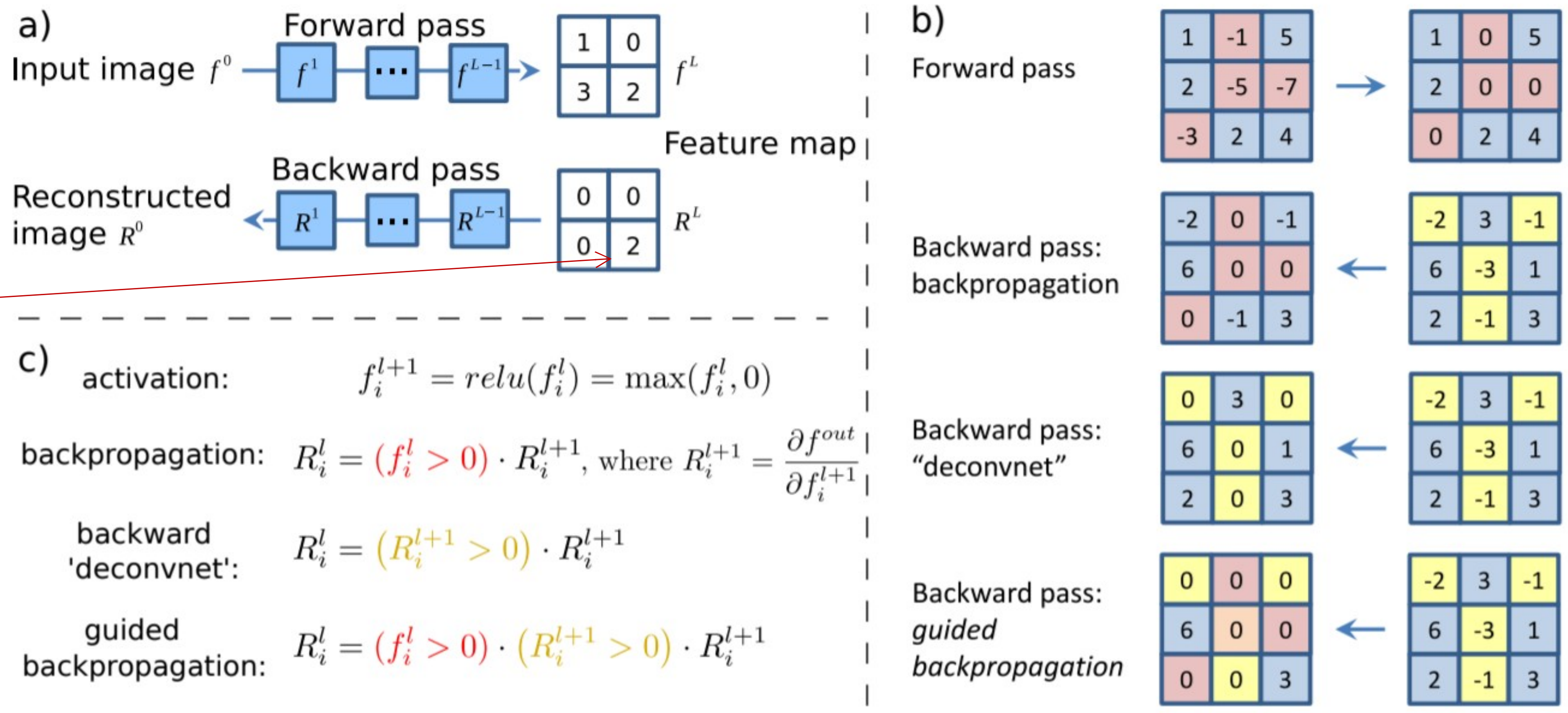
https://www.youtube.com/watch?v=AgkfIQ4IGaM

Activation maps

- Compute the gradient of a given neuron in the network with respect to the input image
  - Select the layer, set the gradient at that level to be all zeros except for the neuron of interest (where we set the gradient at 1)
  - Apply backpropagation to the image

https://arxiv.org/pdf/1412.6806.pdf

Activation maps



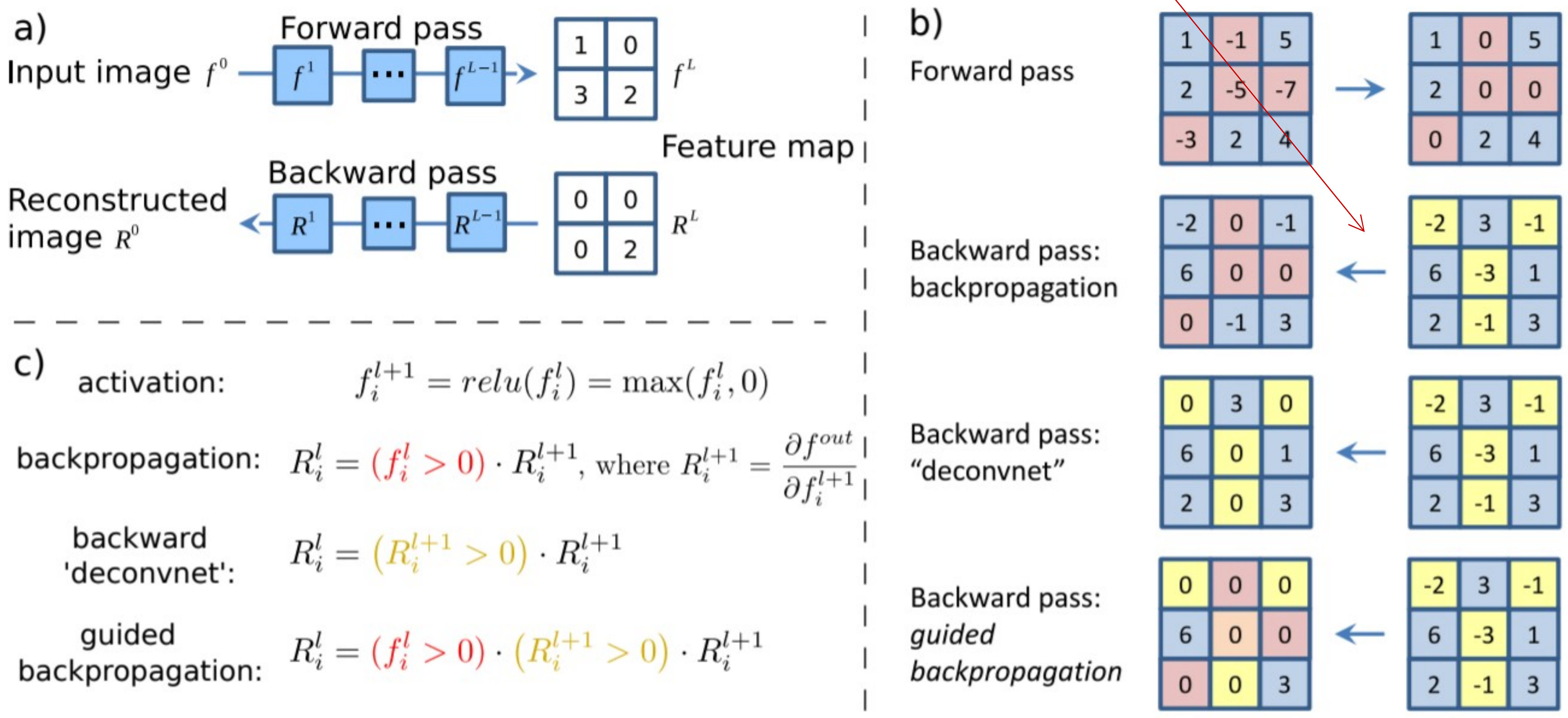
Zero out all the gradients except for the neuron of interest

https://arxiv.org/pdf/1412.6806.pdf

Activation maps

RELU during backward pass: block the gradients for the neurons that had activations < 0

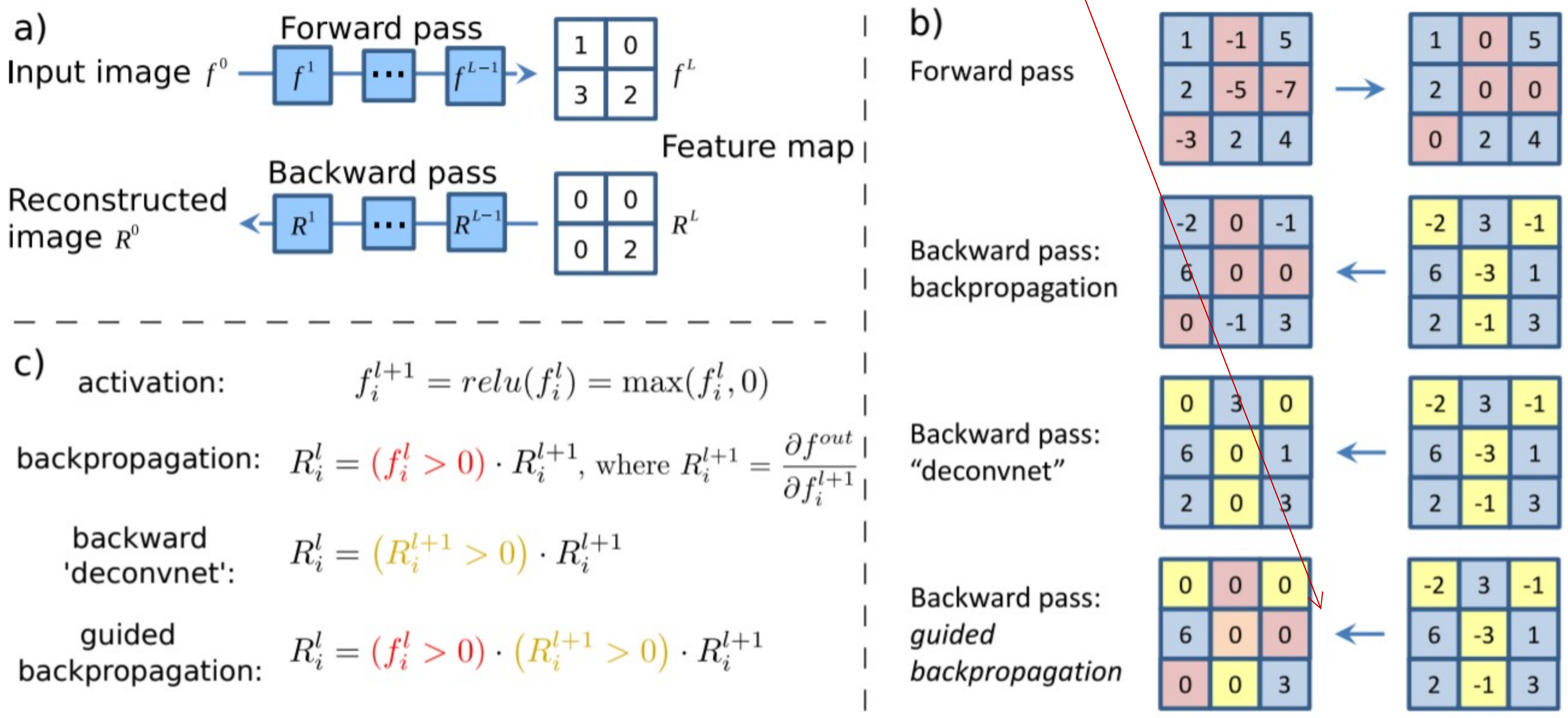
RELU



https://arxiv.org/pdf/1412.6806.pdf

Activation maps

Guided back-propagation: backprop only the parts that have a positive gradient

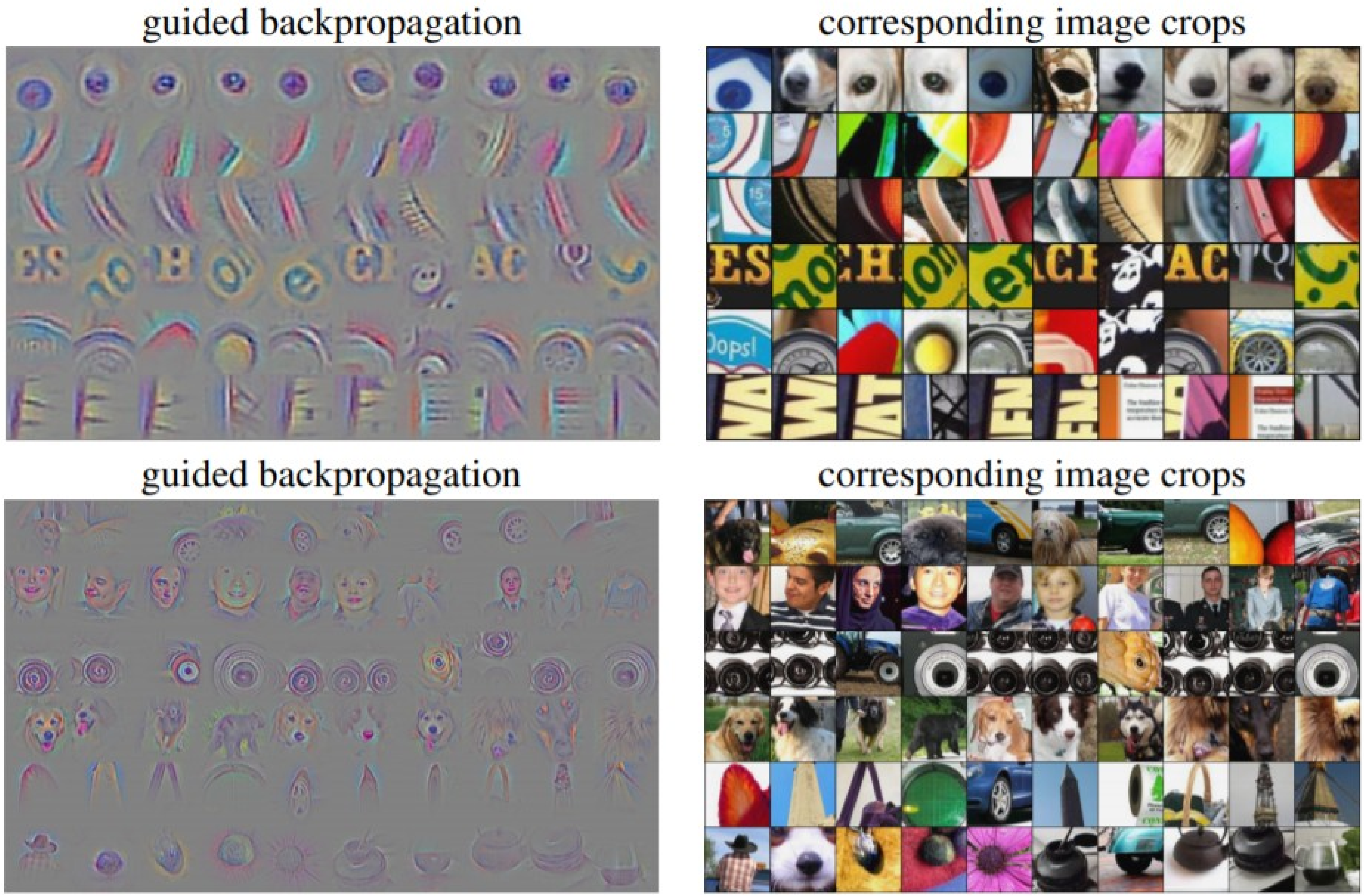


Paths of influence

Pass only gradients that have a positive influence. Keep only the positive influence

https://arxiv.org/pdf/1412.6806.pdf

Activation maps



Visualization of patterns learned by the layer conv6 (top) and layer conv9 (bottom) of the network trained on ImageNet.

Saliency maps – visualize the data gradient

Important image pixels

- Heat map of the gradients
- Compute gradient of the (un-normalized) class score with respect to image pixels and then take absolute value and max over RGB channels
- Strength of influence of each pixel on the class score

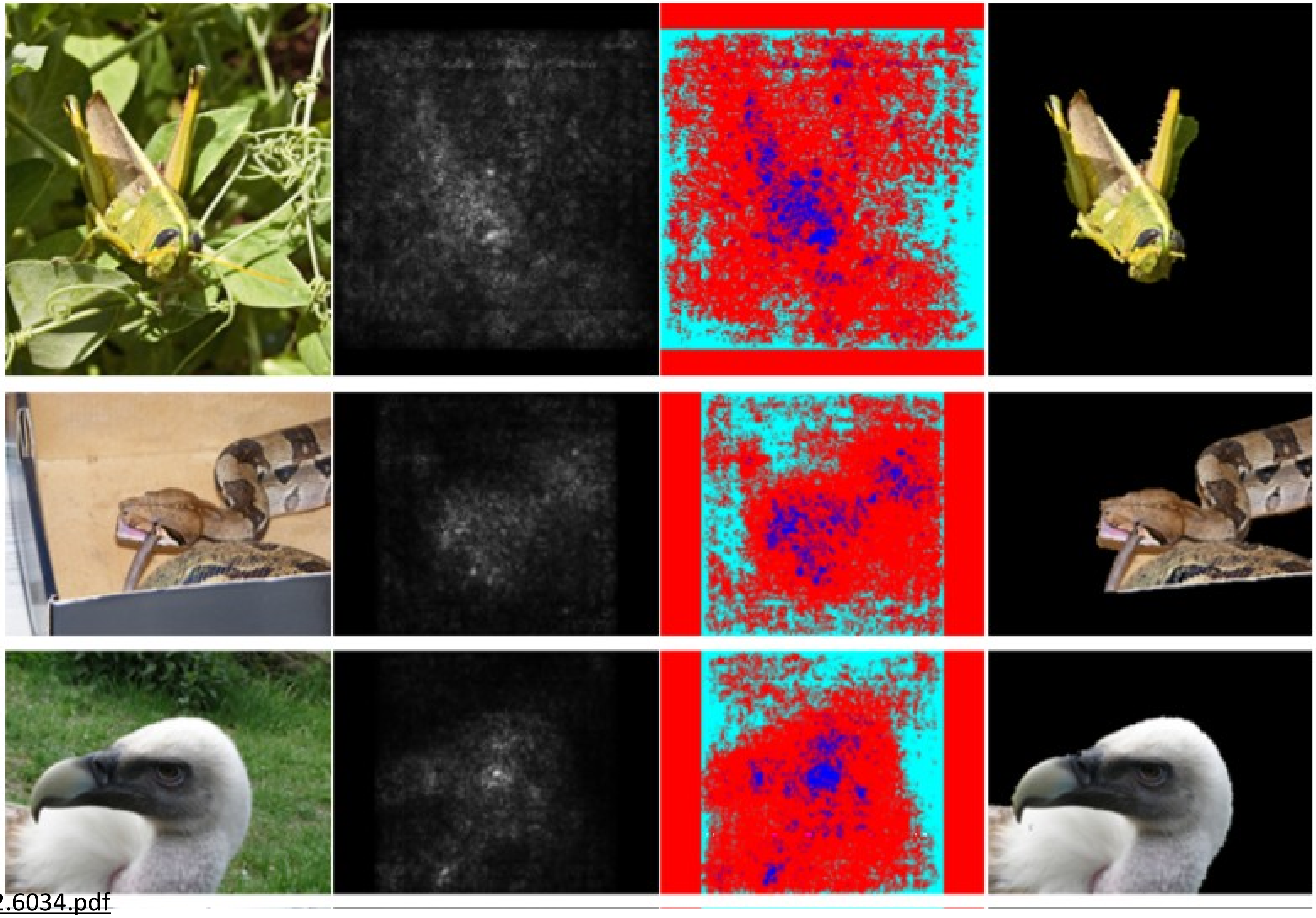
https://arxiv.org/pdf/1312.6034.pdf

Saliency maps – visualize the data gradient

Important image pixels

https://arxiv.org/pdf/1312.6034.pdf

Guided image segmentation

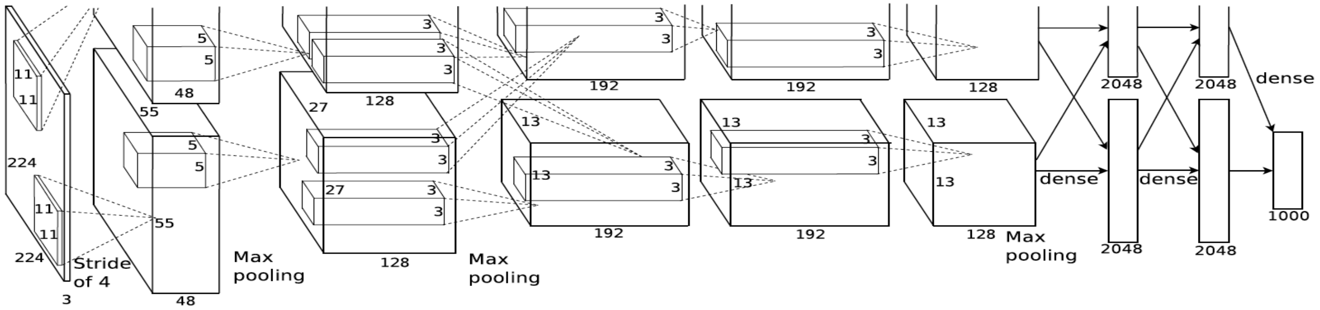


https://arxiv.org/pdf/1312.6034.pdf

Optimize the image

- Keep the parameters of the network fixed and optimize the input image for some class score

Zero image →



Forward pass

Set the gradient of the scores vector to be all zeros except for the class of interest and then back-prop to the image

[0, 0, …., 0, 1, …, 0]

Image update

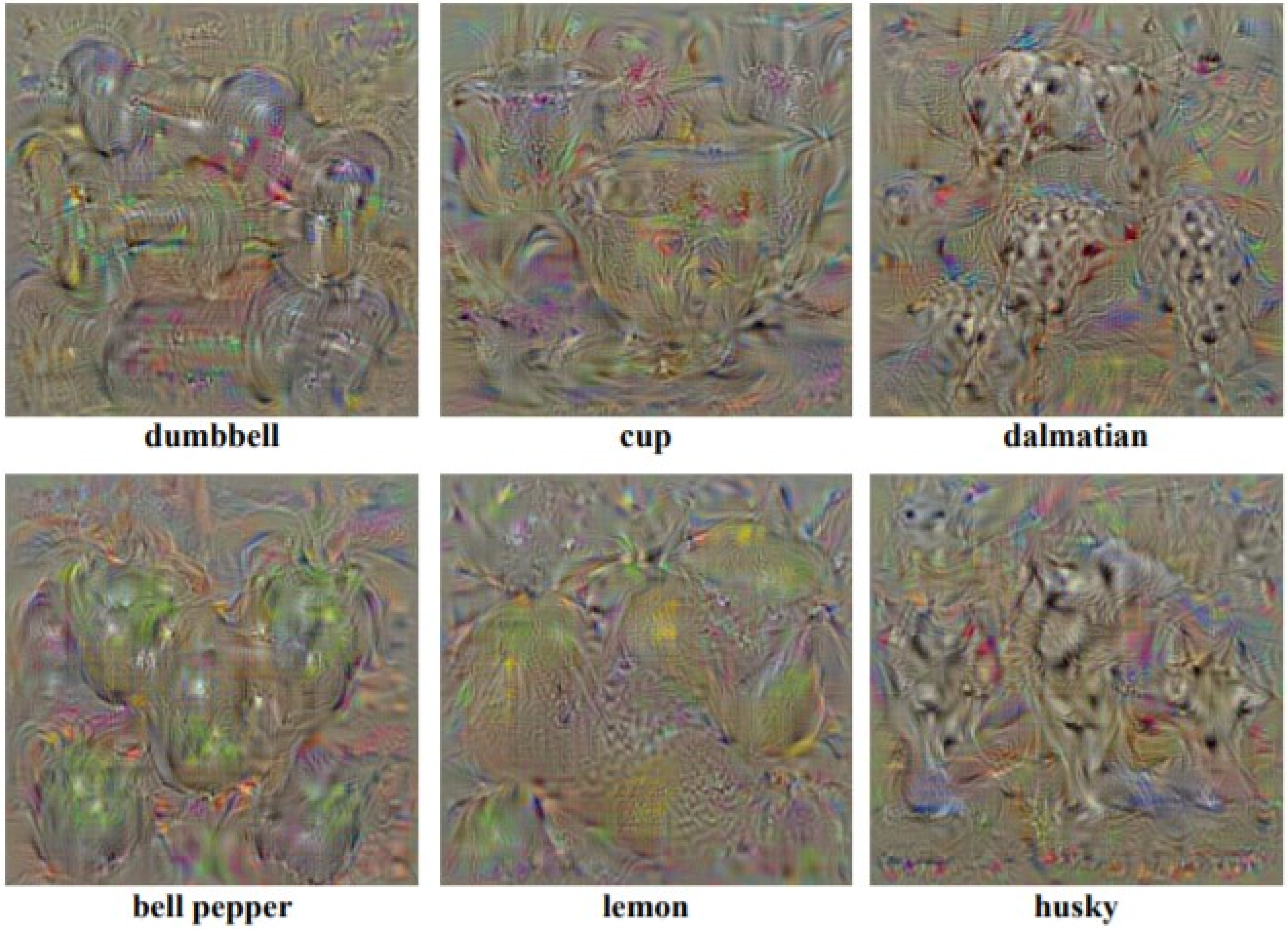
Forward image to the networks

repeat

More formally, let $S_c(I)$ be the score of the class c, computed by the classification layer of the ConvNet for an image I. We would like to find an L_2-regularised image, such that the score S_c is high:

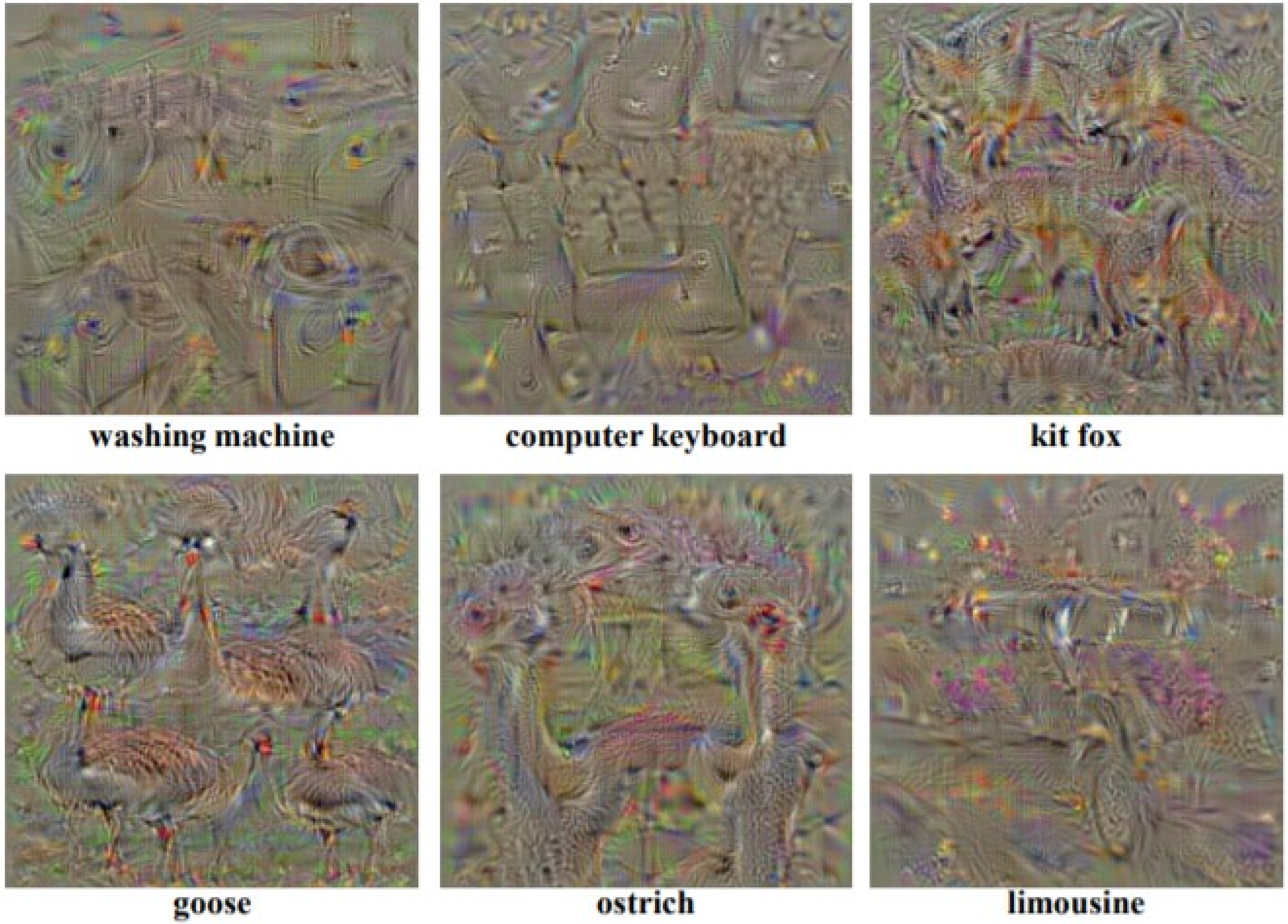
$$\arg\max_I S_c(I) - \lambda \|I\|_2^2, \tag{1}$$

Optimize the image



https://arxiv.org/pdf/1312.6034.pdf

Optimize the image



https://arxiv.org/pdf/1312.6034.pdf

Deep dream

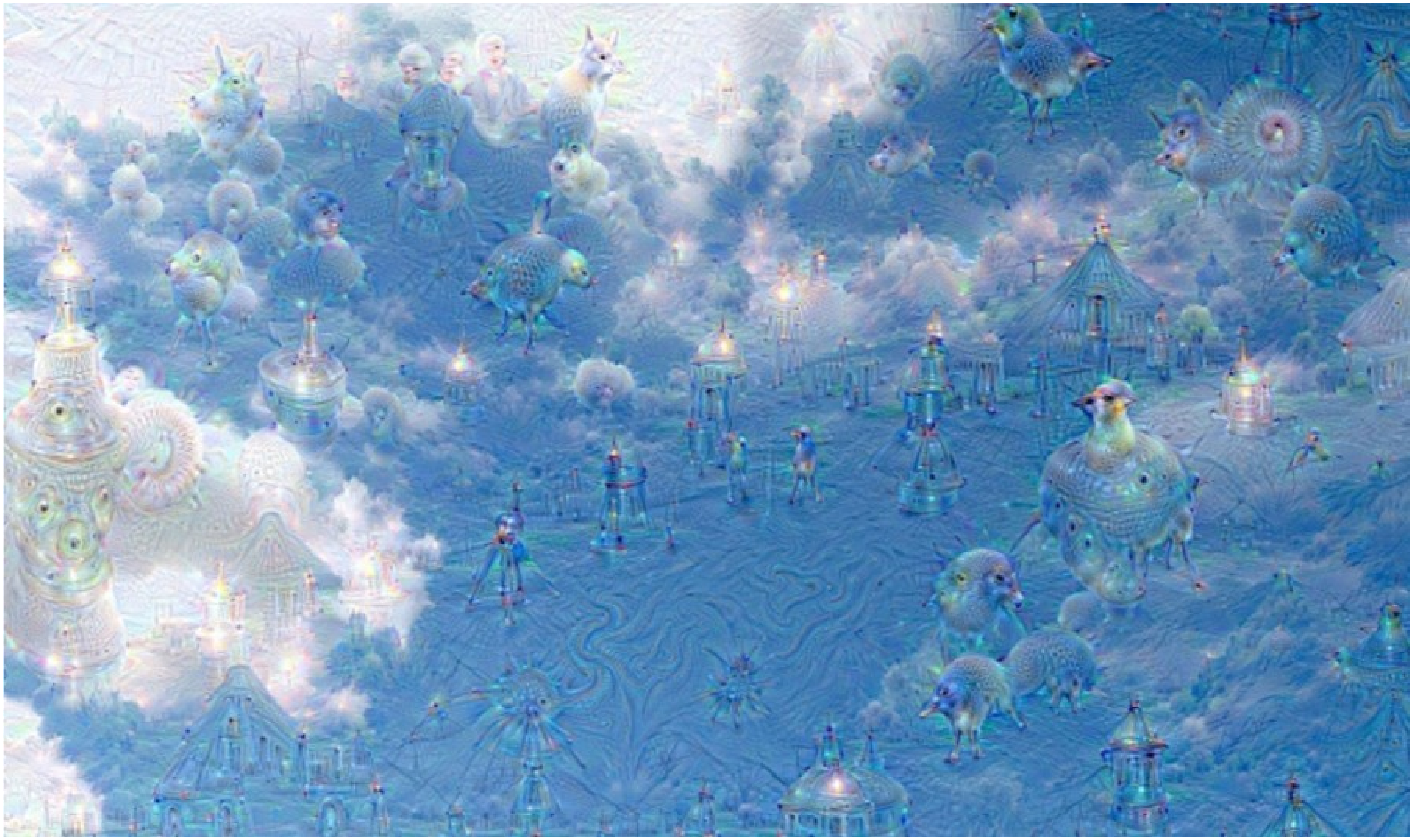
- "Dream" at a given layer of the network (after the ReLu units)
  - Forward pass to the give image
  - Set the gradients to be equal to the activation
  - Backprop to the image

- Modify the image such that it amplifies the activations at any given layer

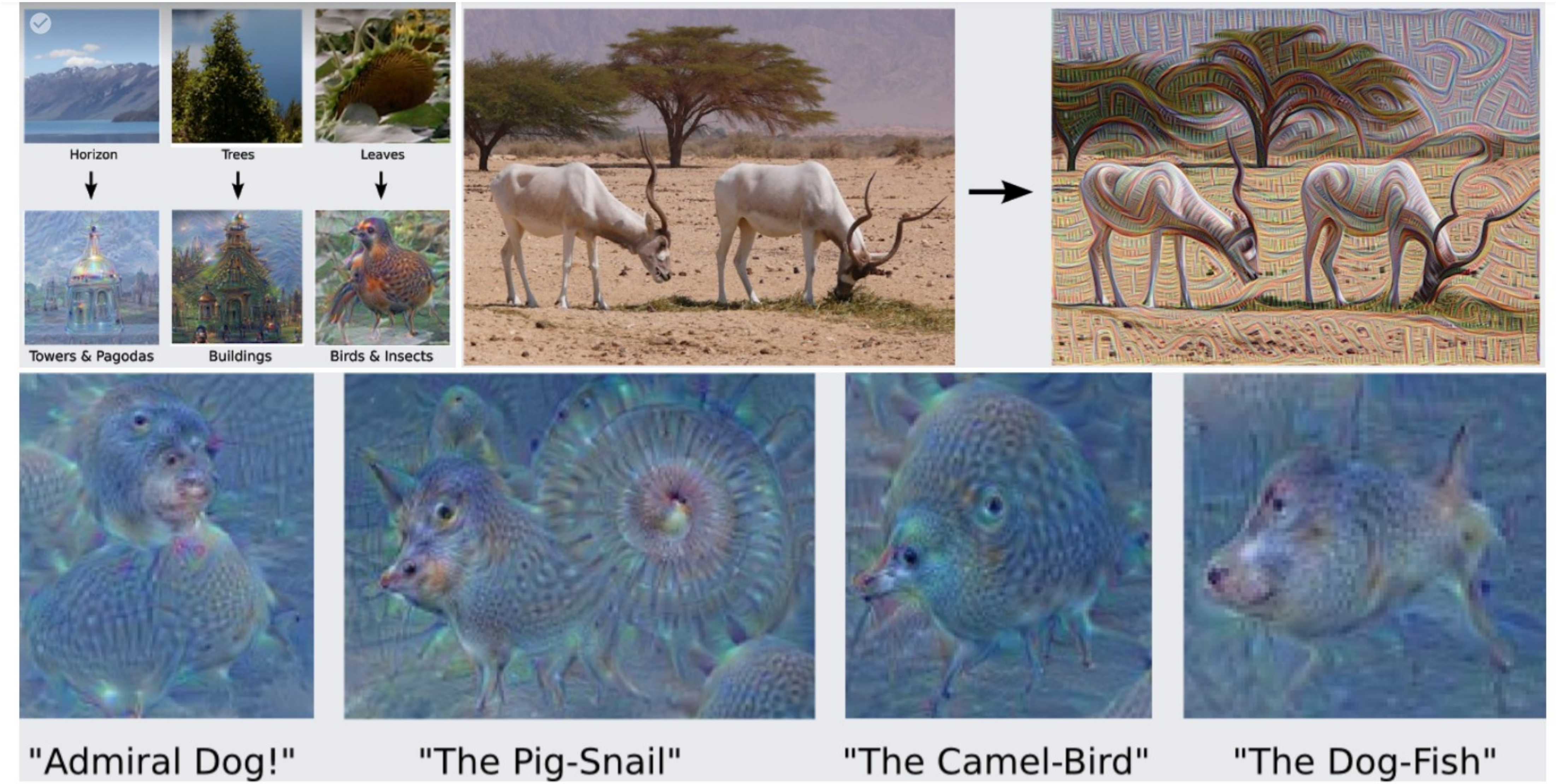
https://github.com/google/deepdream/blob/master/dream.ipynb

Deep dream

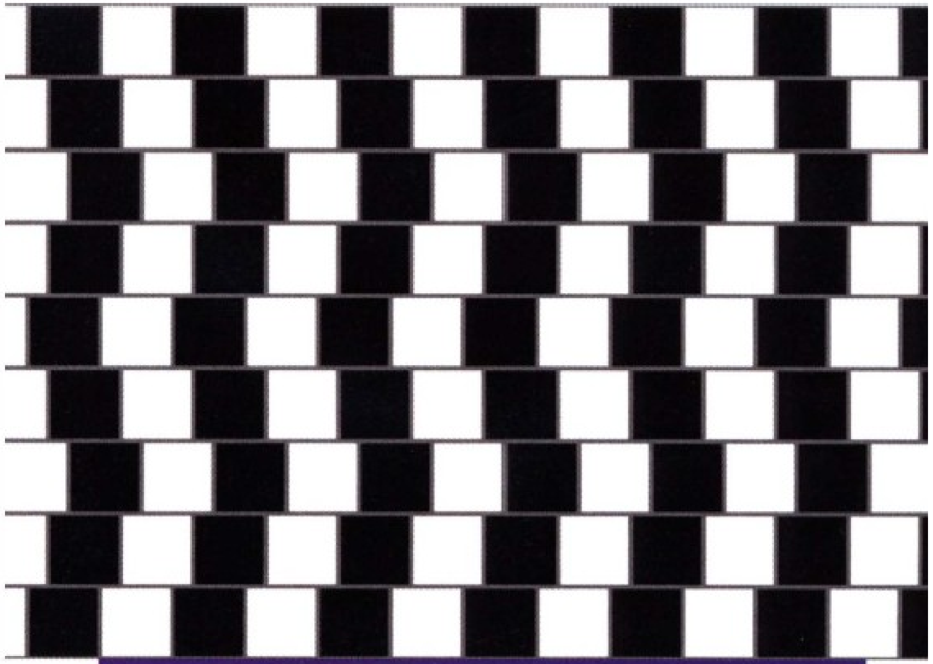
Whatever was activates gets boosted



Deep dream

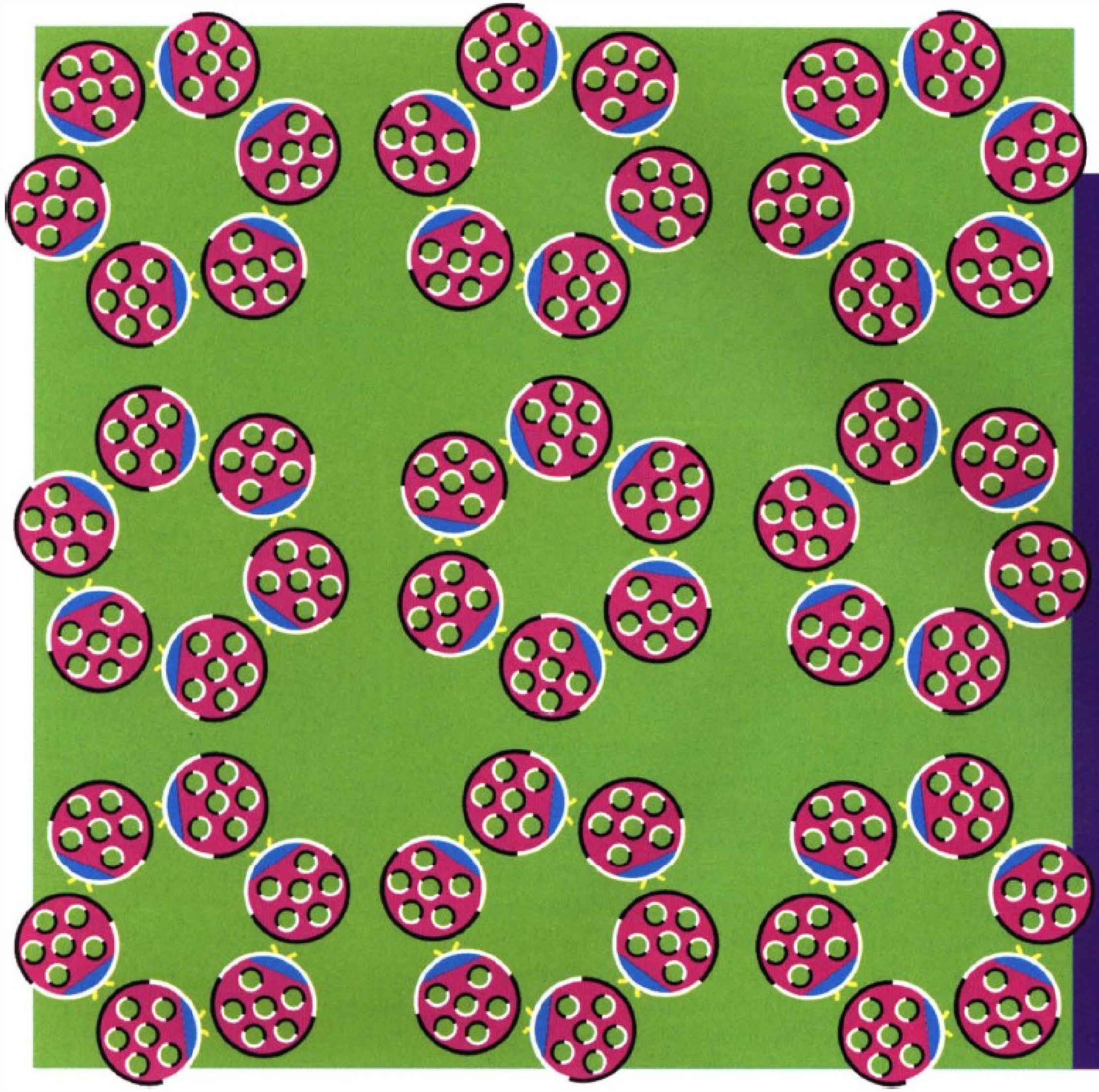

Horizon
Trees
Leaves
Towers & Pagodas
Buildings
Birds & Insects
"Admiral Dog!"
"The Pig-Snail"
"The Camel-Bird"
"The Dog-Fish"

Adversarial examples



Test des rayures

Illusion : les rangées de briques apparaissent en pente alors qu'elles sont pourtant parfaitement parallèles !
Explication : le contraste entre les carrés blancs et noirs avec le léger décalage des carrés noirs d'une rangée à l'autre et la finesse des lignes grises créeraient l'illusion.



Test des coccinelles

Illusion : si on fixe les coccinelles, d'autant plus sur le bord de cette image, elles se mettent à tourner.
Explication : cet effet découle de l'excitation des neurones sensibles au mouvement, majoritaires dans la vision périphérique. Il serait en partie dû au contraste entre des couleurs claires à l'avant de la coccinelle et des couleurs sombres à l'arrière, car ces informations ne parviennent pas simultanément aux aires visuelles. Le cerveau interpréterait alors le mouvement comme un déroulement temporel. L'effet est accentué par les "microsaccades" naturelles des yeux.

“Adversarial” examples for the human brain

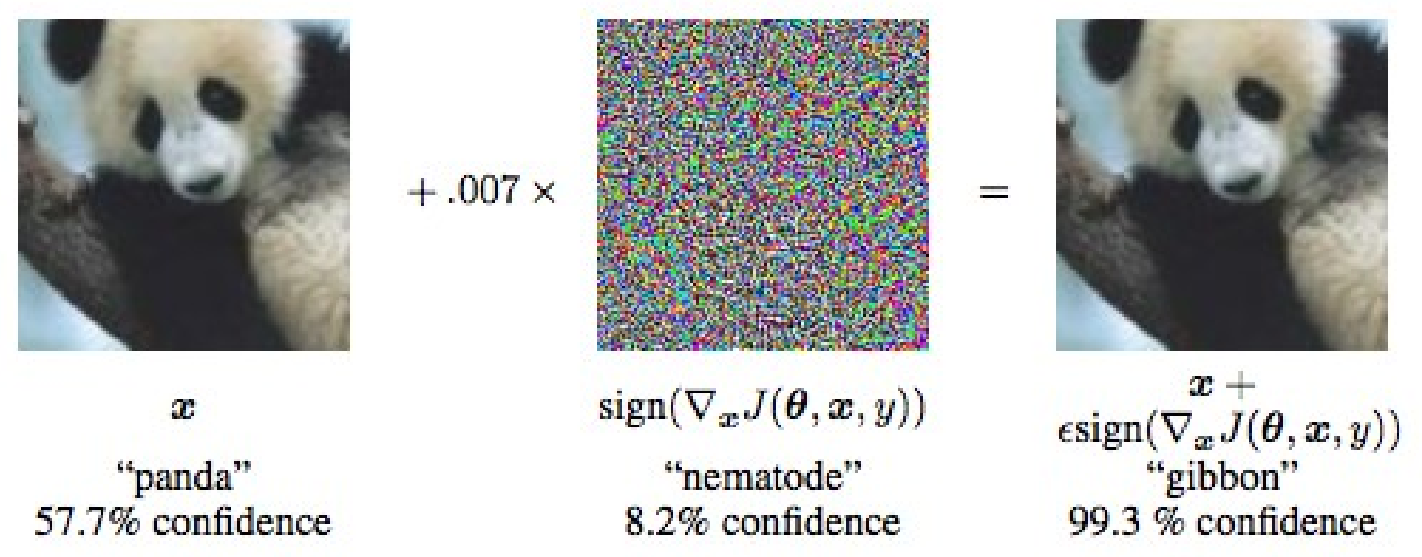
#thedress

#whiteandgold

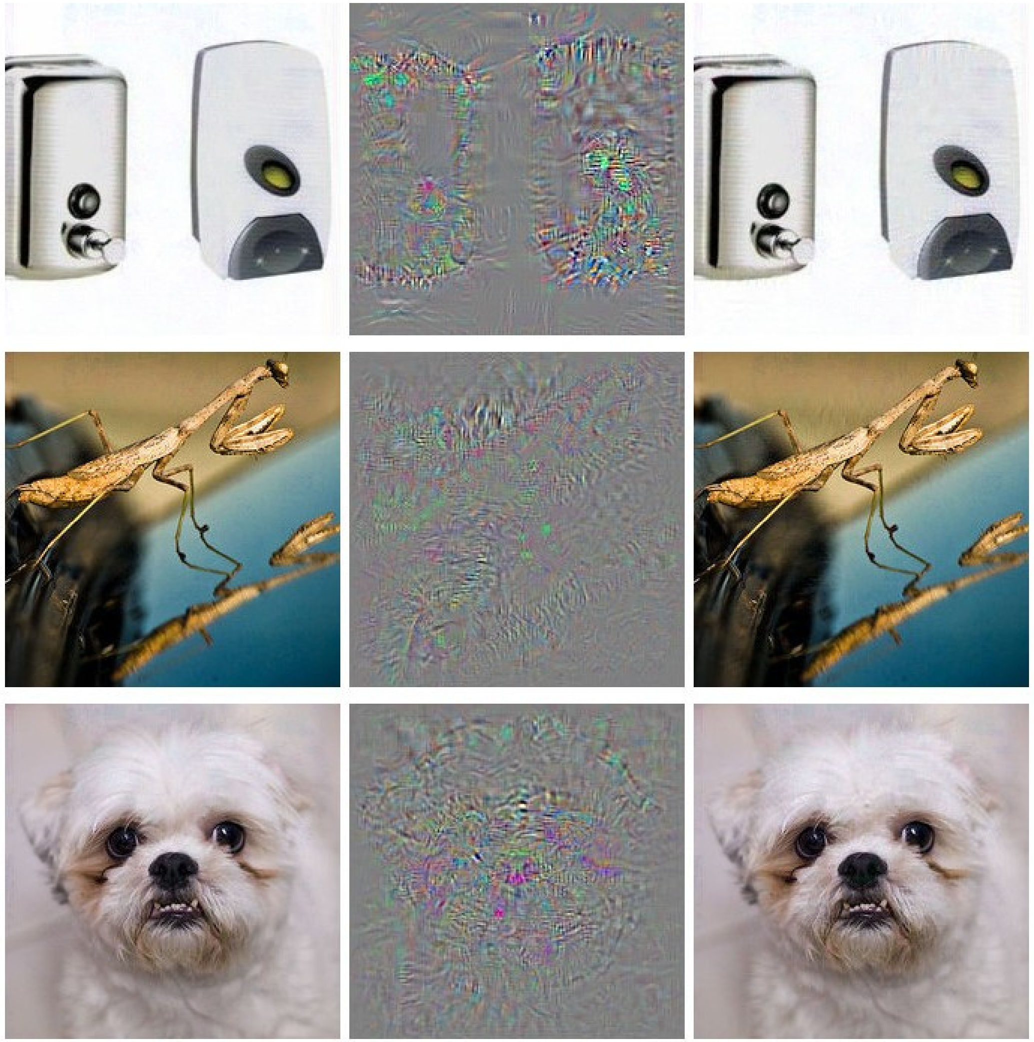
#blackandblue

Adversarial examples

An adversarial example is an instance with small, intentional feature perturbations that cause a machine learning model (**!not only CNNs!**) to make a false prediction.

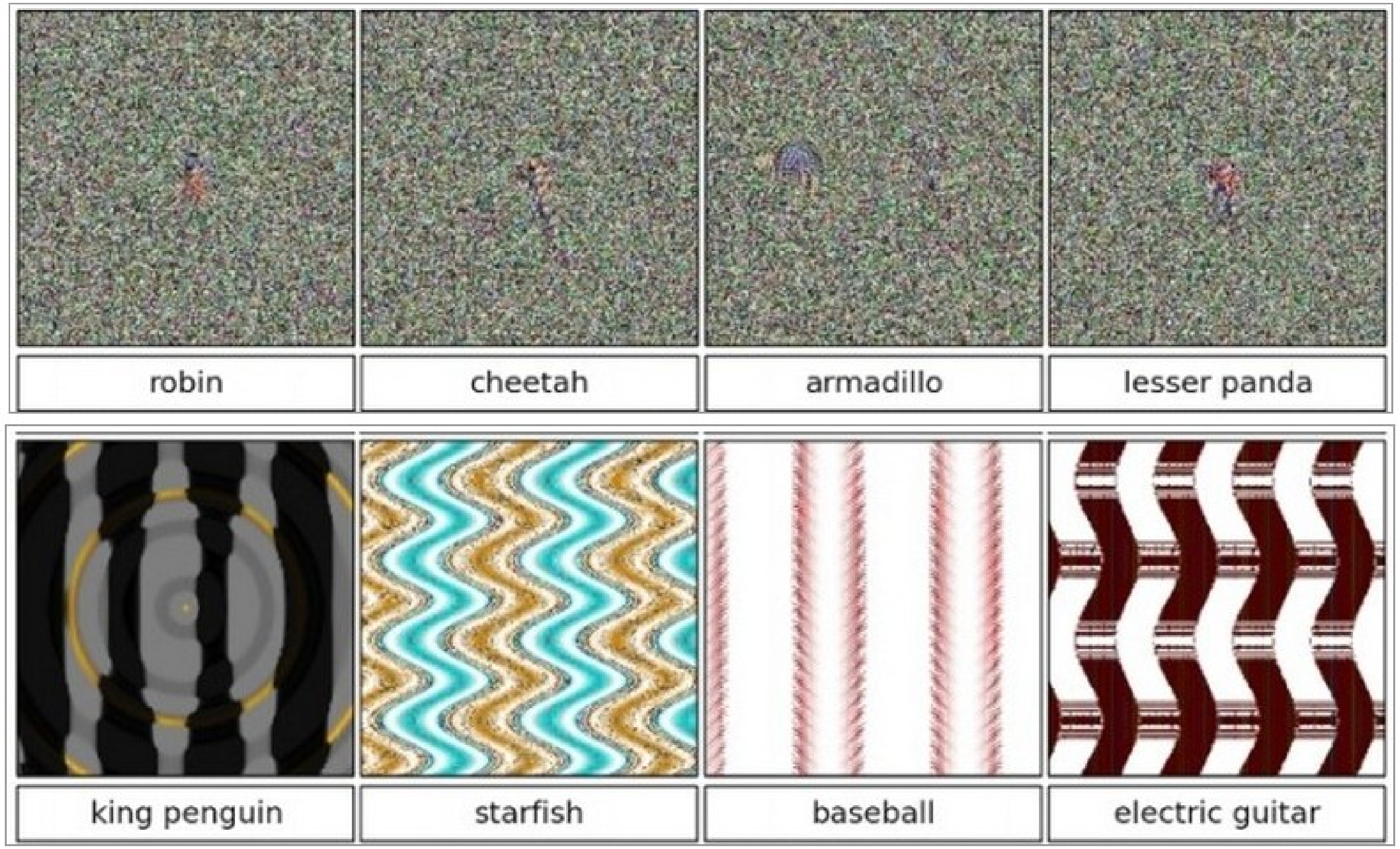


Make everything an ostrich



Szegedy, Christian, et al. "Intriguing properties of neural networks." *arXiv preprint arXiv:1312.6199* (2013).

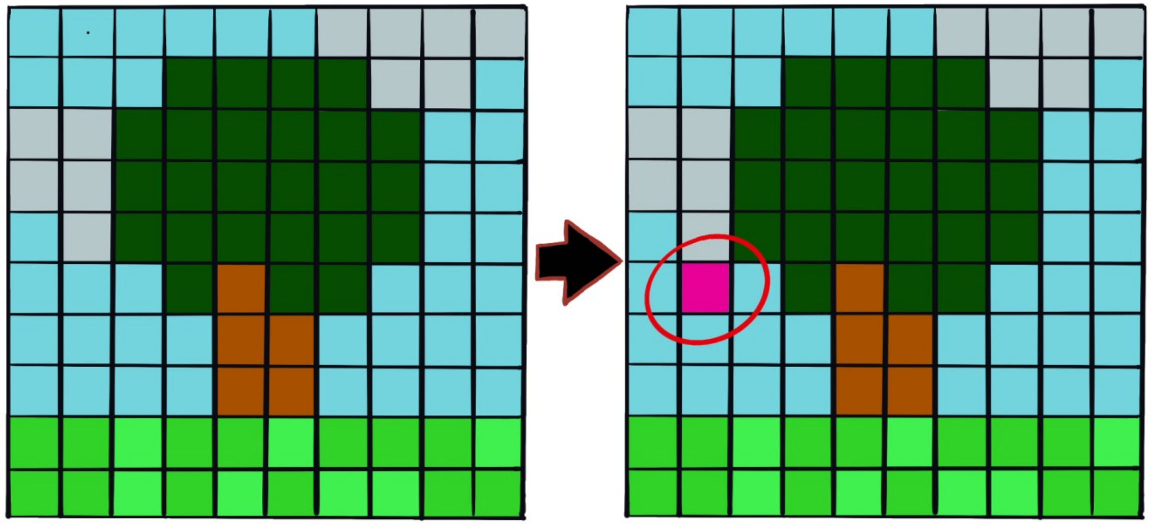
Classifying noise



Nguyen, Anh, Jason Yosinski, and Jeff Clune. "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

1-pixel attacks

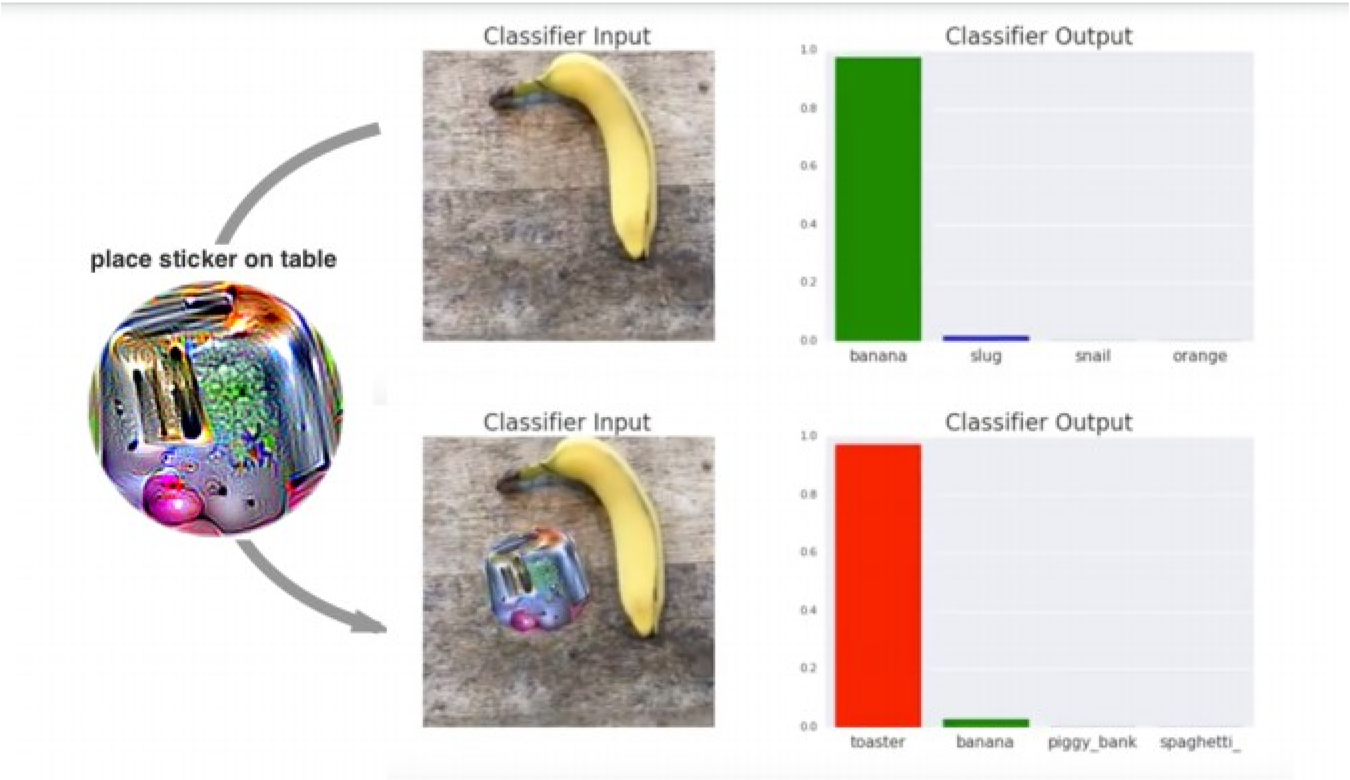
Constraint: when designing the adversarial example only one pixel may change



One Pixel Attack for Fooling Deep Neural Networks, Su et al. , 2019

Adversarial patch

Make everything a toaster



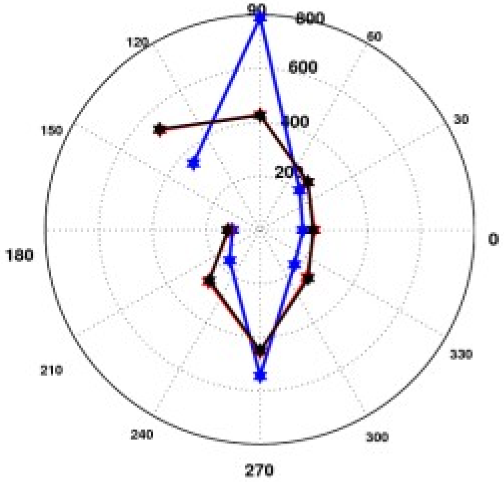
Slight alterations in the physical world

https://spectrum.ieee.org/cars-that-think/transportation/sensors/slight-street-sign-modifications-can-fool-machine-learning-algorithms

Adversarial examples

occur not only for CNNs but also for handcrafted features

HOG descriptor



Exploring the Representation Capabilities of the HOG Descriptor, Tatu et. al