#DLUPC

Day 3 Lecture 4

Interpretability

Eva Mohedano
eva.mohedano@insight-centre.org

Postdoctoral Researcher
Insight-centre for Data Analytics
Dublin City University

DCU

Insight Centre for Data Analytics

http://bit.ly/dlcv2018

Acknowledgements

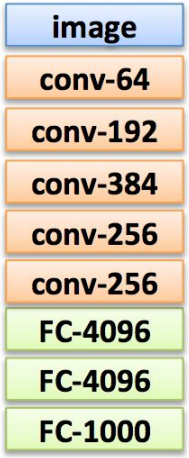
[Amaia Salvador](mailto:amaia.salvador@upc.edu)
amaia.salvador@upc.edu

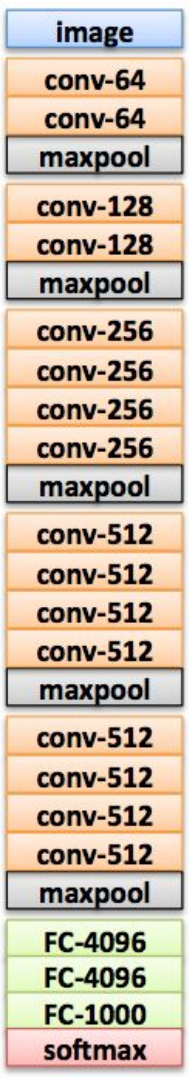
PhD Candidate
Universitat Politècnica de Catalunya

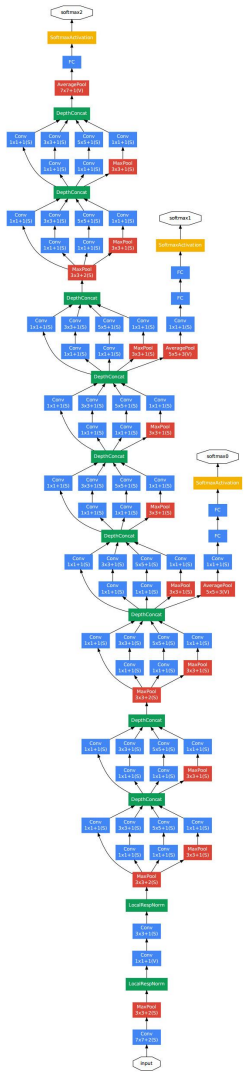
UNIVERSITAT POLITÈCNICA DE CATALUNYA

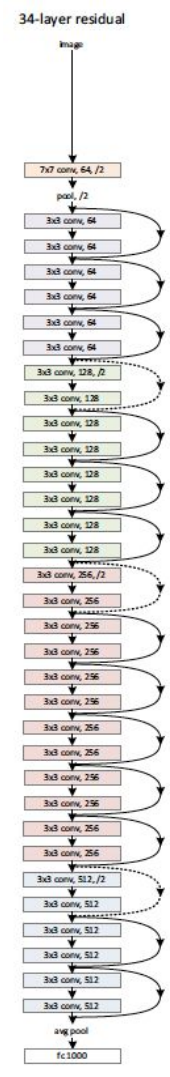
[UPC DLCV 2016]

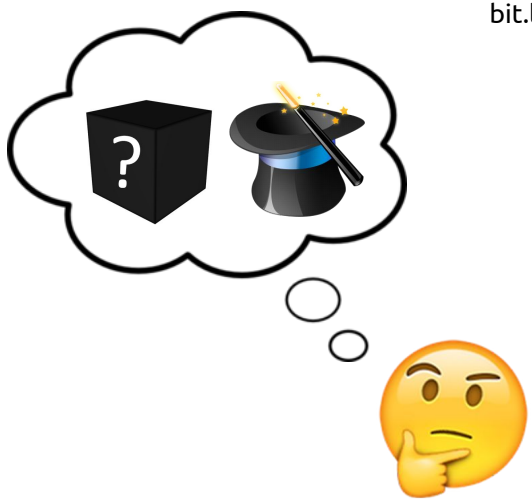
AlexNet



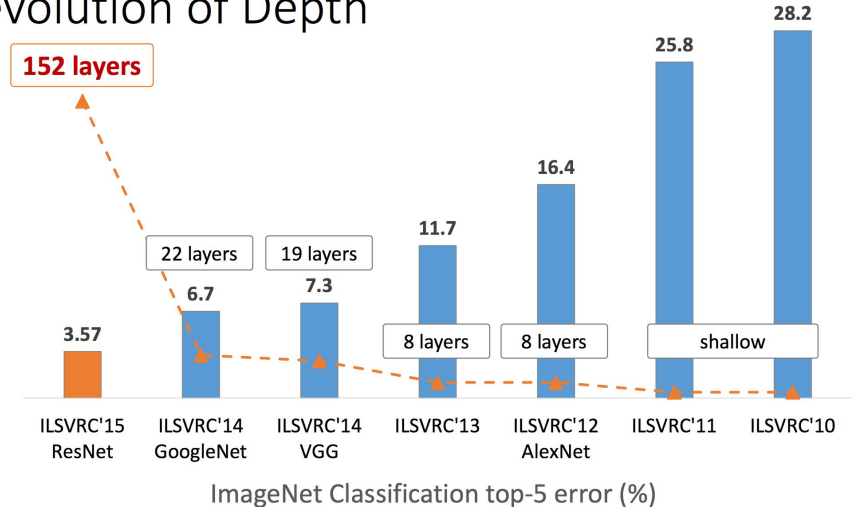








Revolution of Depth



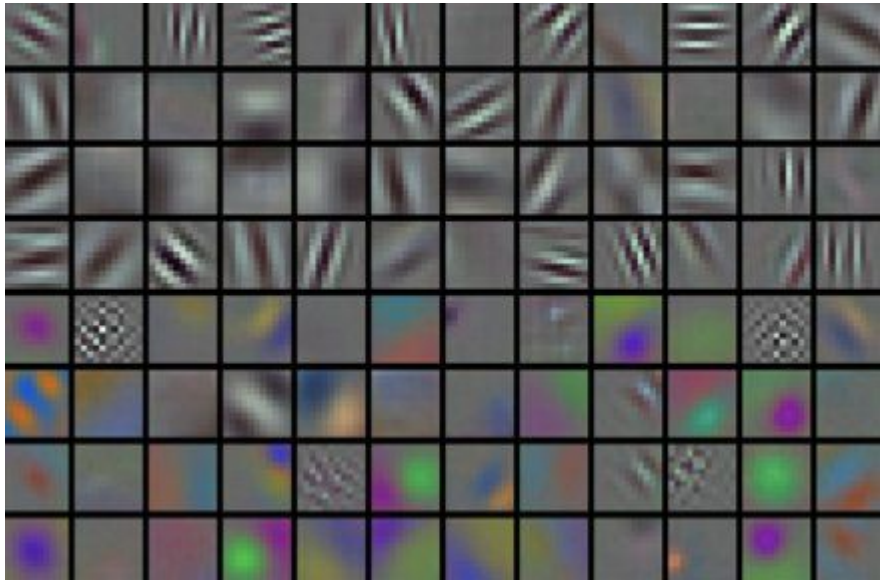
Overview

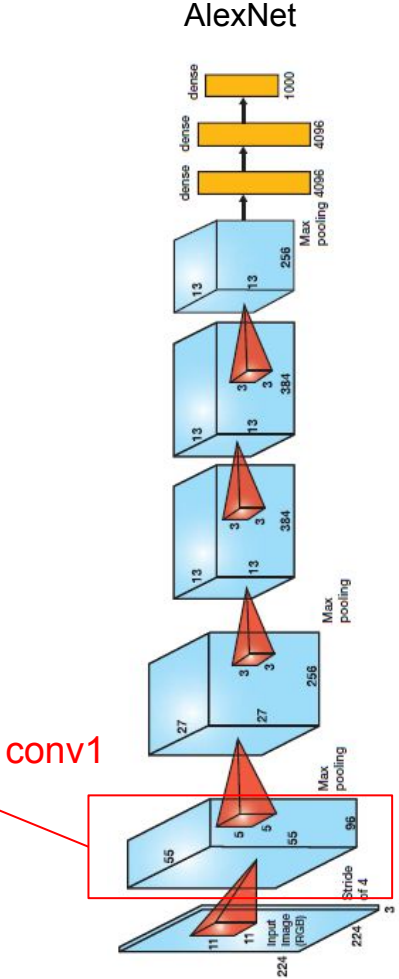
- Learned weights
- Activations from data
- Gradient-based
- Activation-maximization (AM)

Overview

- **Learned weights**
- Activations from data
- Gradient-based
- Activation-maximization (AM)

Visualize Learned Weights





Only convolutional filters from first layer can be “visualized”, because their depth of matches the input RGB channels.

Visualize Learned Weights (demo)

Filters with depth larger than 3 can be visualized showing a gray-scale image of each depth, one next to the other.

layer 2 weights

Source: ConvnetJS

Visualize Learned Weights (demo)

Demo: Classify MNIST digits with a Convolutional Neural Network

ConvNetJS

Deep Learning in your browser

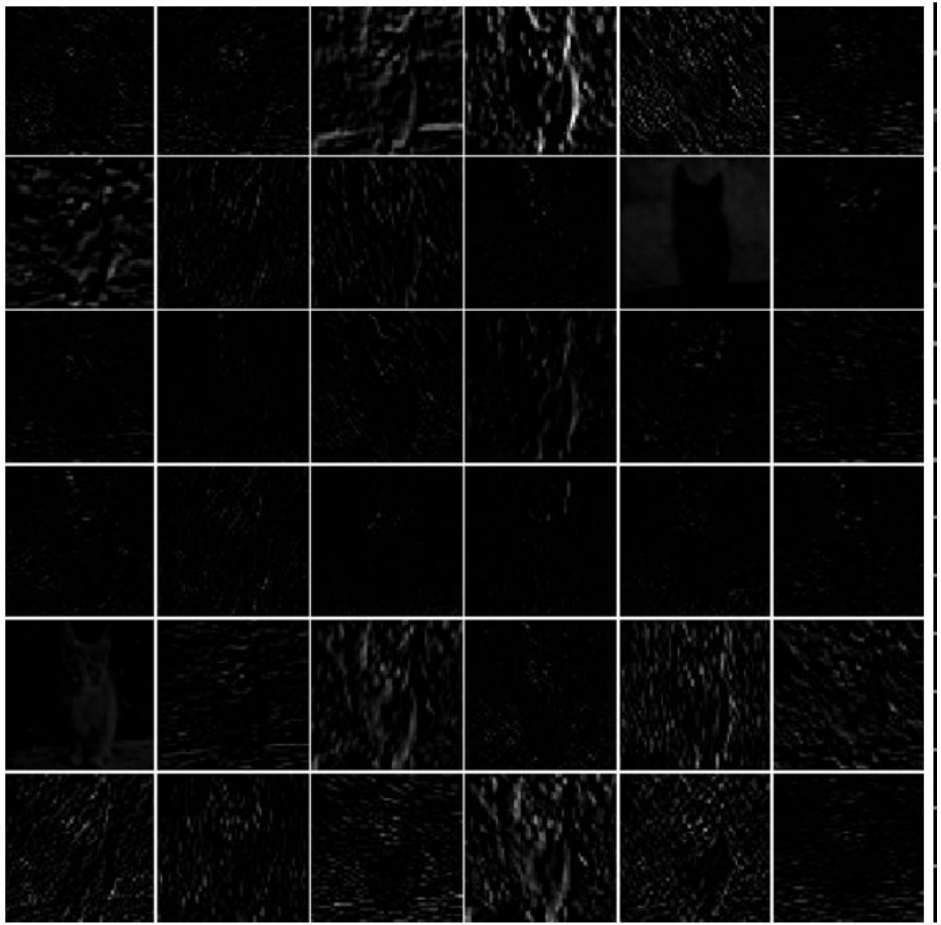
“ConvNetJS is a Javascript library for training Deep Learning models (mainly Neural Networks) entirely in your browser. Open a tab and you're training. No software requirements, no compilers, no installations, no GPUs, no sweat.”

Overview

- Learned weights
- **Activations from data**
- Gradient-based
- Activation-maximization (AM)

Activations from data

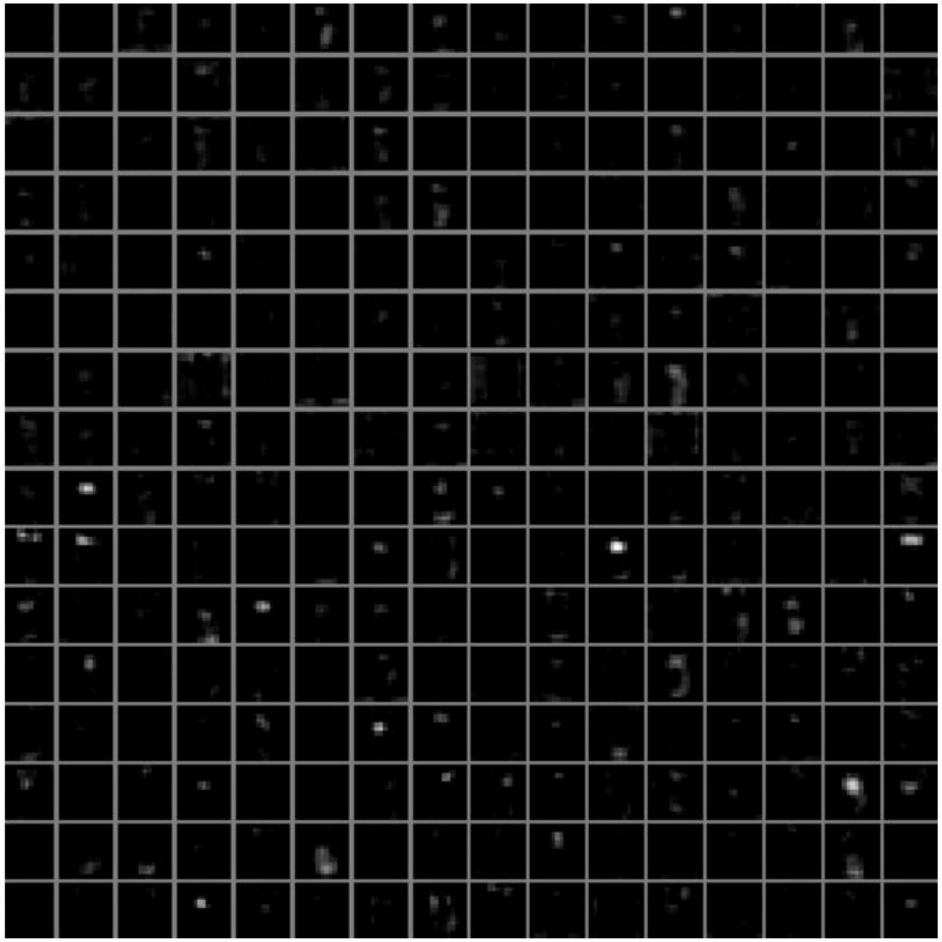
Input
image



conv1

Activations from data

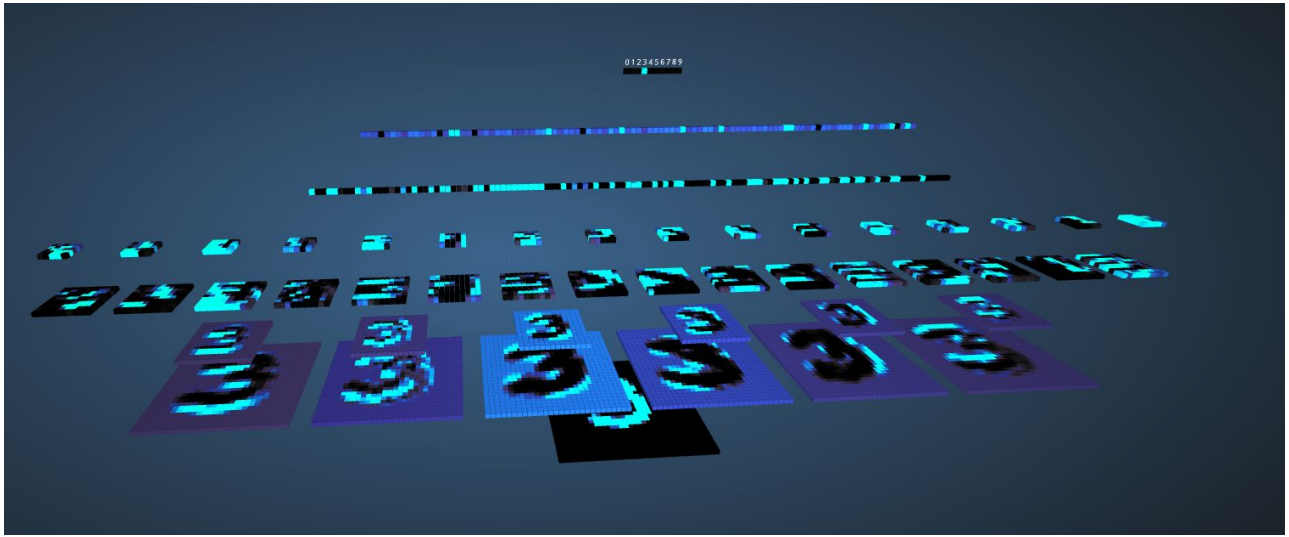
Input
image



conv5

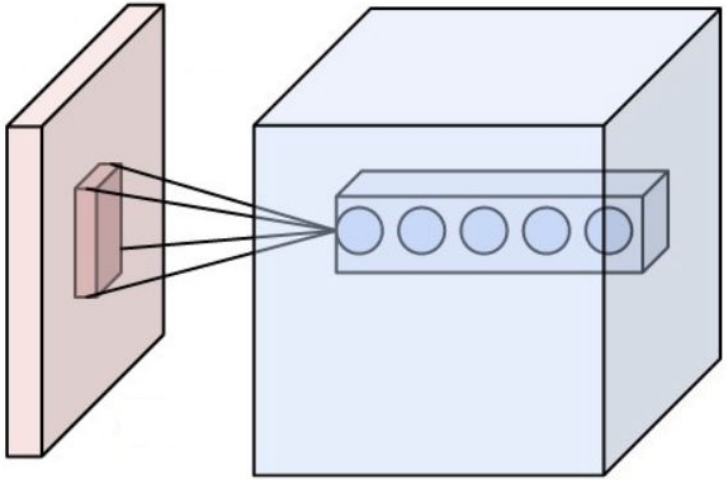
Activations from data

Demo: 3D Visualization of a Convolutional Neural Network



Harley, Adam W. "An Interactive Node-Link Visualization of Convolutional Neural Networks." In Advances in Visual Computing, pp. 867-877. Springer International Publishing, 2015.

Reminder: Receptive Field



Receptive field: Part of the input that is visible to a neuron. It increases as we stack more convolutional layers (i.e. neurons in deeper layers have larger receptive fields).

Receptive Field

Visualize the receptive field of a neuron on those images that activate it the most

“people”

“text”

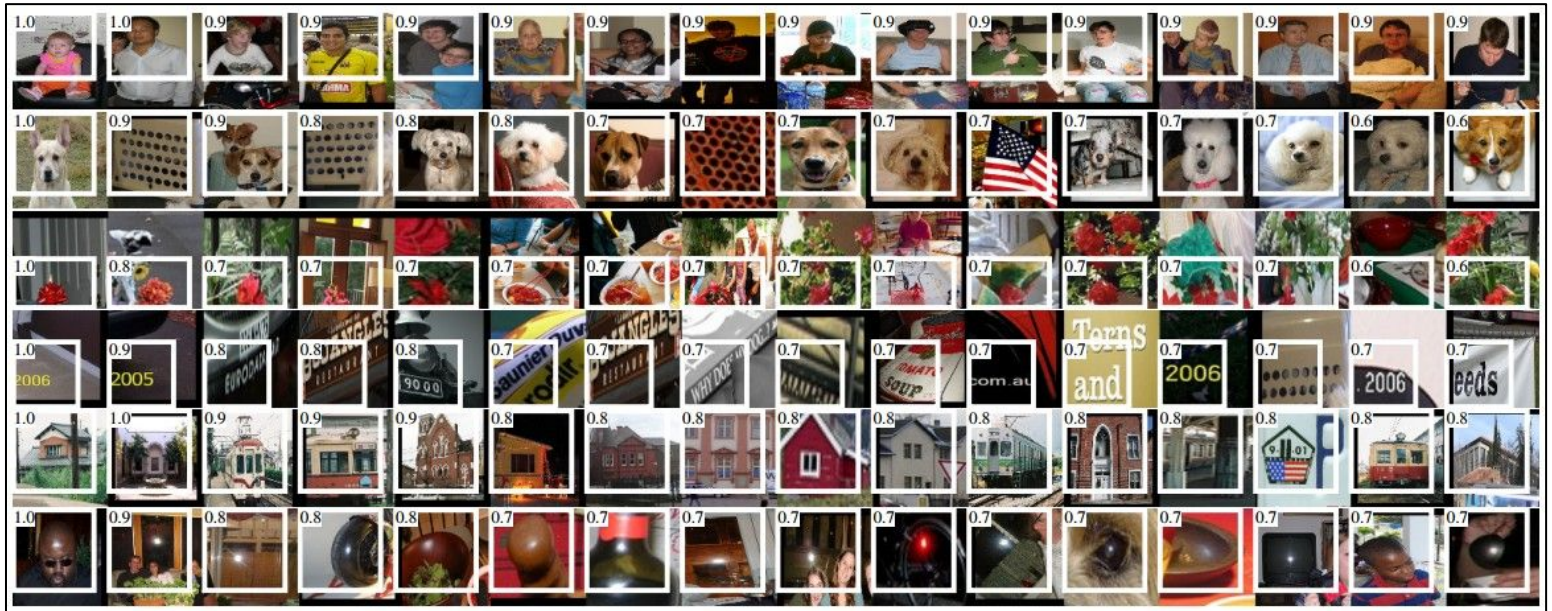


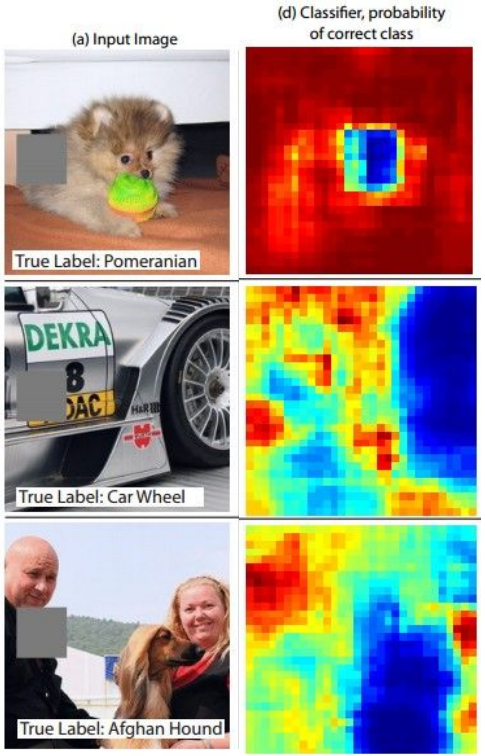
Figure 4: Top regions for six pool$_5$ units. Receptive fields and activation values are drawn in white. Some units are aligned to concepts, such as people (row 1) or text (4). Other units capture texture and material properties, such as dot arrays (2) and specular reflections (6).

Girshick et al. Rich feature hierarchies for accurate object detection and semantic segmentation. CVPR 2014

Occlusion experiments

1. Iteratively forward the same image through the network, occluding a different region at a time.

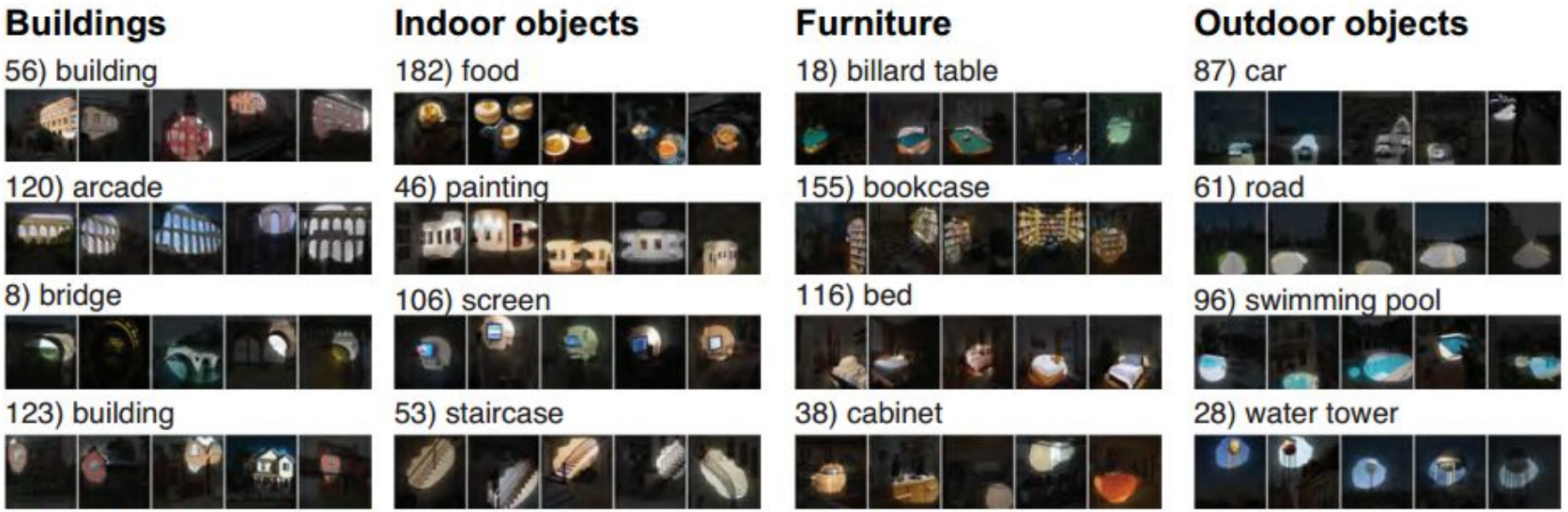
2. Keep track of the probability of the correct class w.r.t. the position of the occluder



Zeiler and Fergus. Visualizing and Understanding Convolutional Networks. ECCV 2015

Occlusion experiments

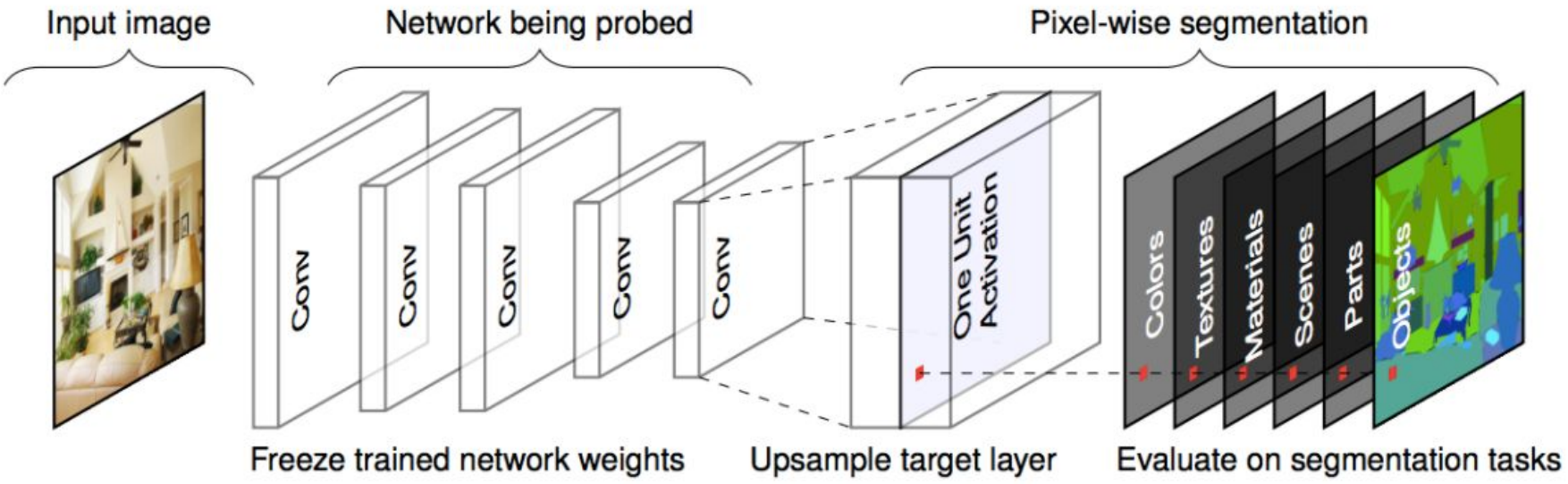
The changes in activations can be observed in any layer. This has allowed identifying some filters as object detectors.



Zhou, Bolei, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. "Object detectors emerge in deep scene CNNs." ICLR 2015.

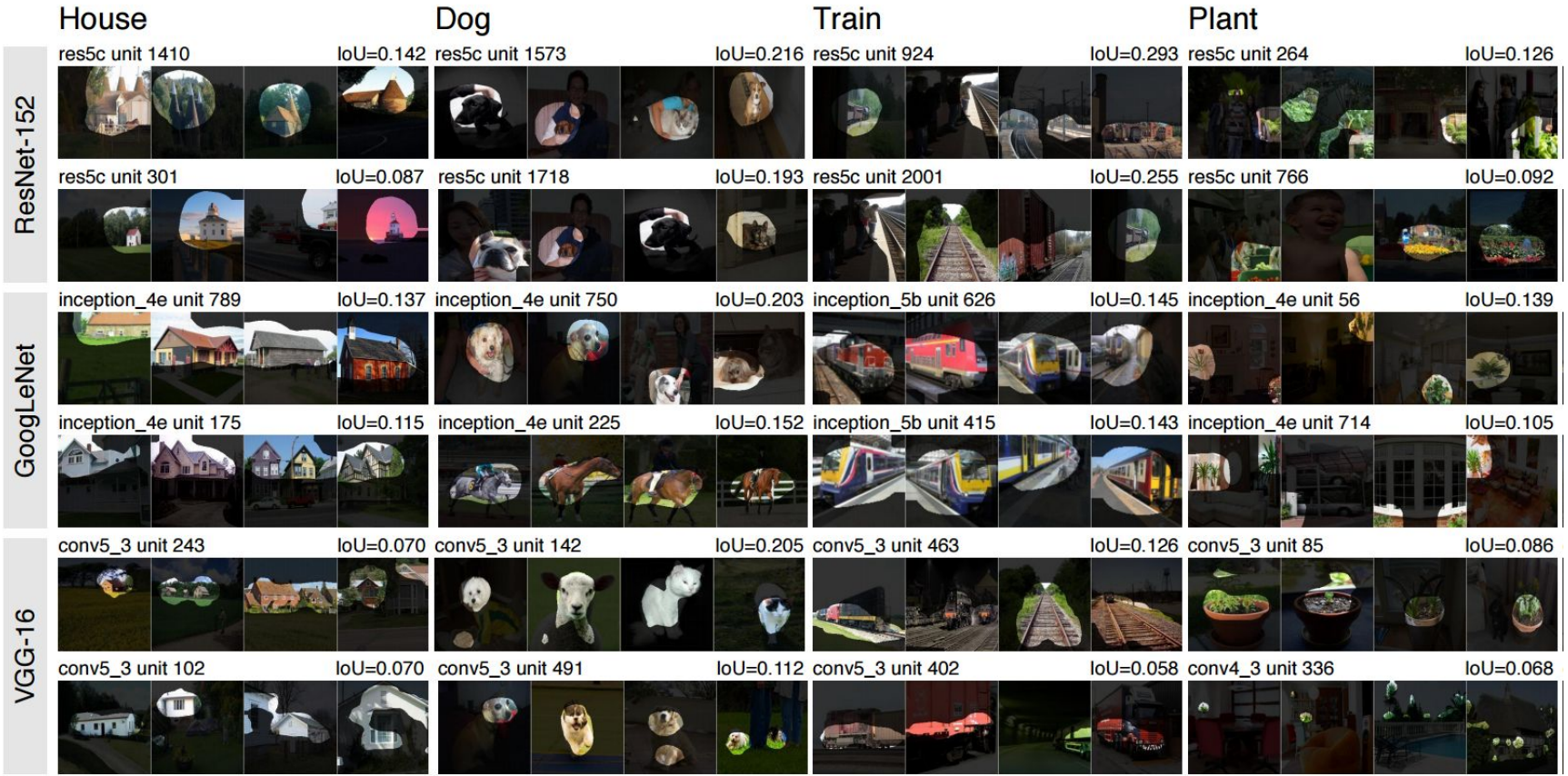
Network Dissection

Same idea, but automatic unit labeling using densely labeled dataset (pixel-level annotations).
The thresholded activation of each conv unit in the network is evaluated for semantic segmentation.



Bau, Zhou et al. Network Dissection: Quantifying Interpretability of Deep Visual Representations. CVPR 2017

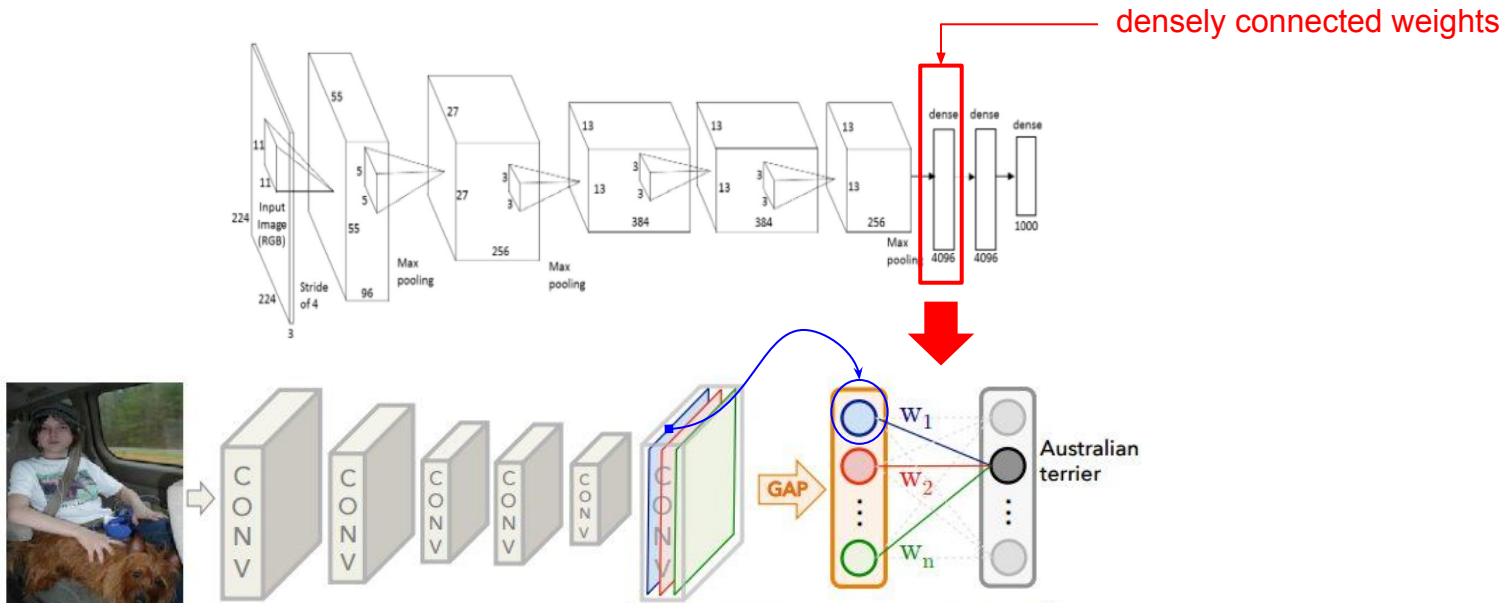
Network Dissection



Bau, Zhou et al. Network Dissection: Quantifying Interpretability of Deep Visual Representations. CVPR 2017

Class Activation Maps

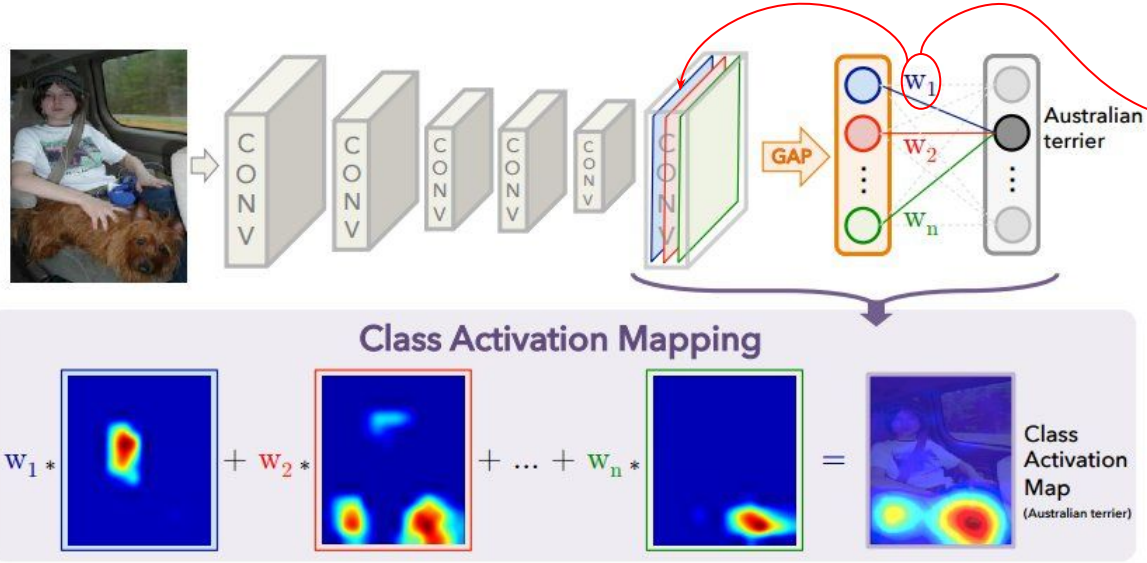
Class Activation Maps (CAMs): Replace FC layer after last conv with Global Average Pooling (GAP).



Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. "Learning deep features for discriminative localization." CVPR 2016

Class Activation Maps

Class Activation Maps (CAMs): The classifier weights define the contribution of each channel in the previous layer

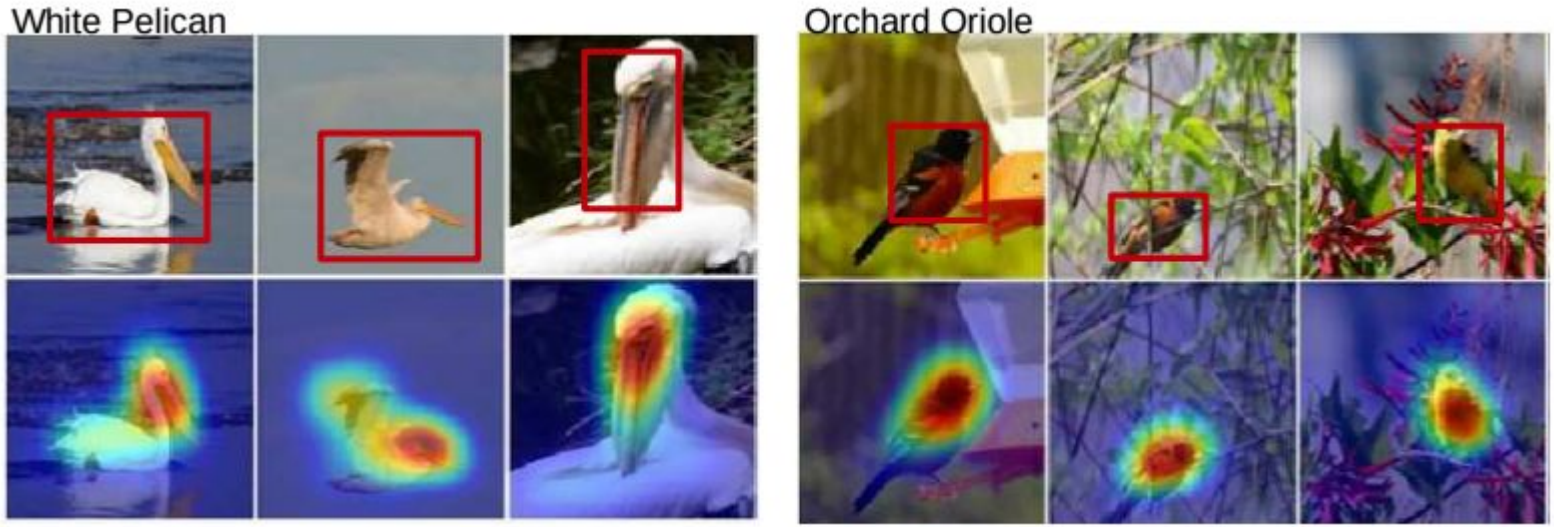


contribution of the *blue* channel in the conv layer to the prediction of "australian terrier"

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. "Learning deep features for discriminative localization." CVPR 2016

Class Activation Maps

Class Activation Maps (CAMs): Unsupervised object localization



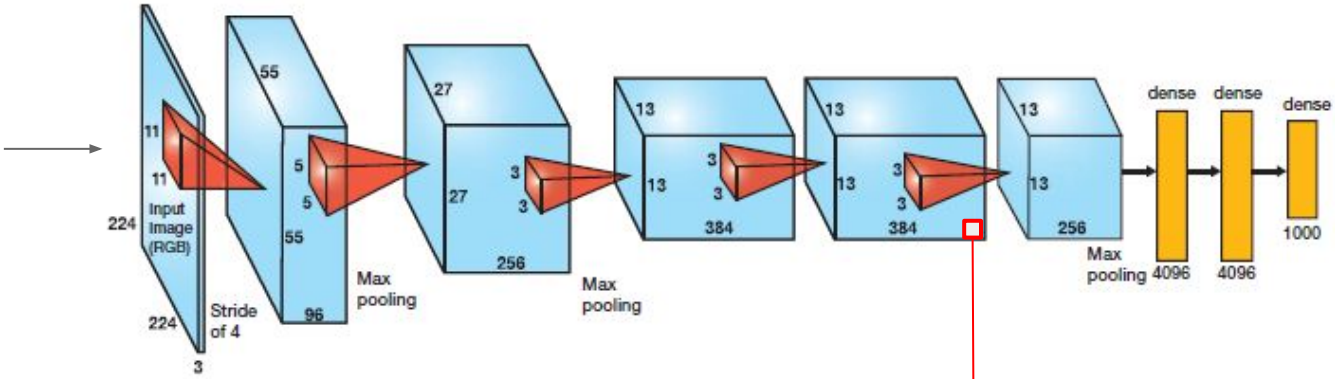
Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. "Learning deep features for discriminative localization." CVPR 2016

Overview

- Learned weights
- Activations from data
- **Gradient-based**
- Activation-maximization (AM)

Gradient-based approach

Goal: Visualize the part of an image that mostly activates one of the neurons.



Compute the gradient of any neuron w.r.t. the image

1. Forward image up to the desired layer (e.g. conv5)
2. Set all gradients to 0
3. Set gradient for the neuron we are interested in to 1
4. Backpropagate to get reconstructed image (gradient on the image)

Gradient-based approach

Goal: Visualize the part of an image that mostly activates one of the neurons.

Springenberg, Dosovitskiy, et al. Striving for Simplicity: The All Convolutional Net. ICLR 2015

Gradient-based approach

guided backpropagation

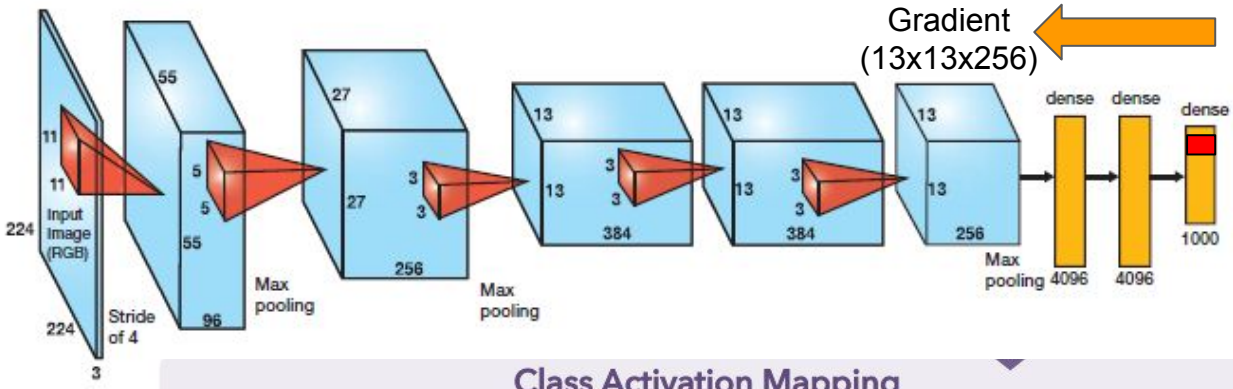
corresponding image crops

Springenberg, Dosovitskiy, et al. Striving for Simplicity: The All Convolutional Net. ICLR 2015

Grad-CAM

Same idea as Class Activation Maps (CAMs), but:

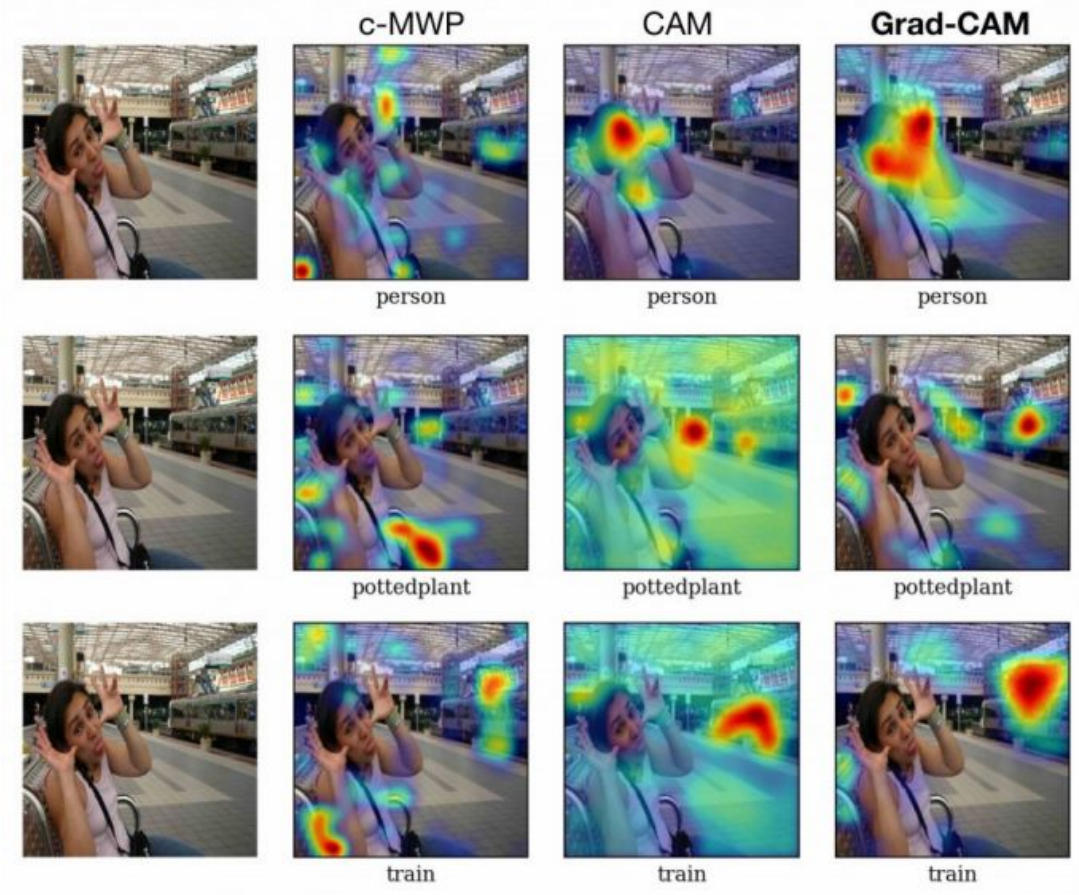
- No modifications required to the network
- Weight feature map by the gradient of the class wrt each channel



Feature maps weighted by the mean gradient.

Ramprasaath et al. Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization. ICCV 2017

Grad-CAM



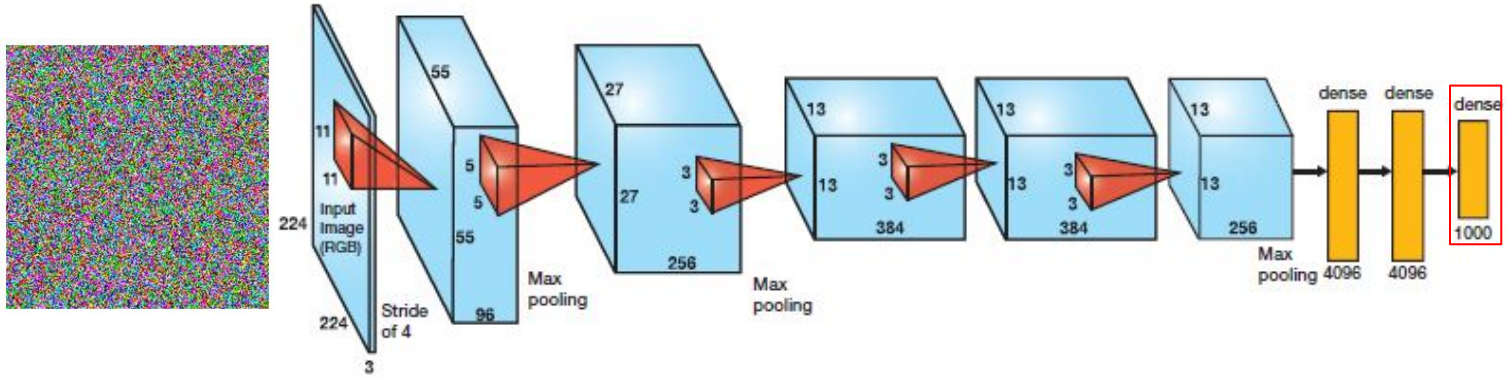
Ramprasaath et al. Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization. ICCV 2017

Overview

- Learned weights
- Activations from data
- Gradient-based
- **Activation Maximization (AM)**

Activation Maximization (AM)

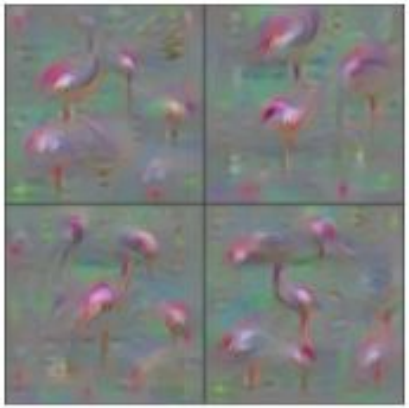
Goal: Generate an image that maximizes the activation of a neuron (or class confidence).

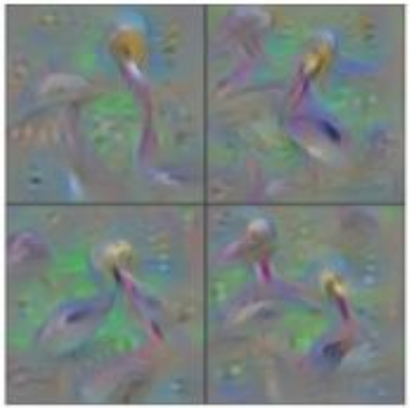


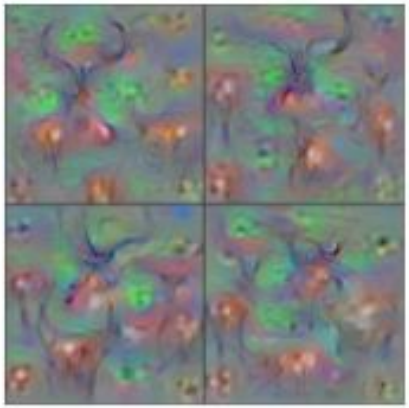
1. Forward random image
2. Set the gradient of the confidence vector to be [0,0,0…,1,...,0,0]
3. Backprop to get gradient on the image
4. Update image (small step in the gradient direction)
5. Repeat

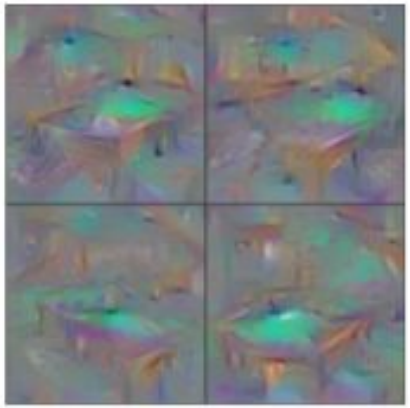
K. Simonyan, A. Vedaldi, A. Ziseramnn, Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, ICLR 2014

Activation Maximization (AM)


Flamingo


Pelican


Hartebeest


Billiard Table

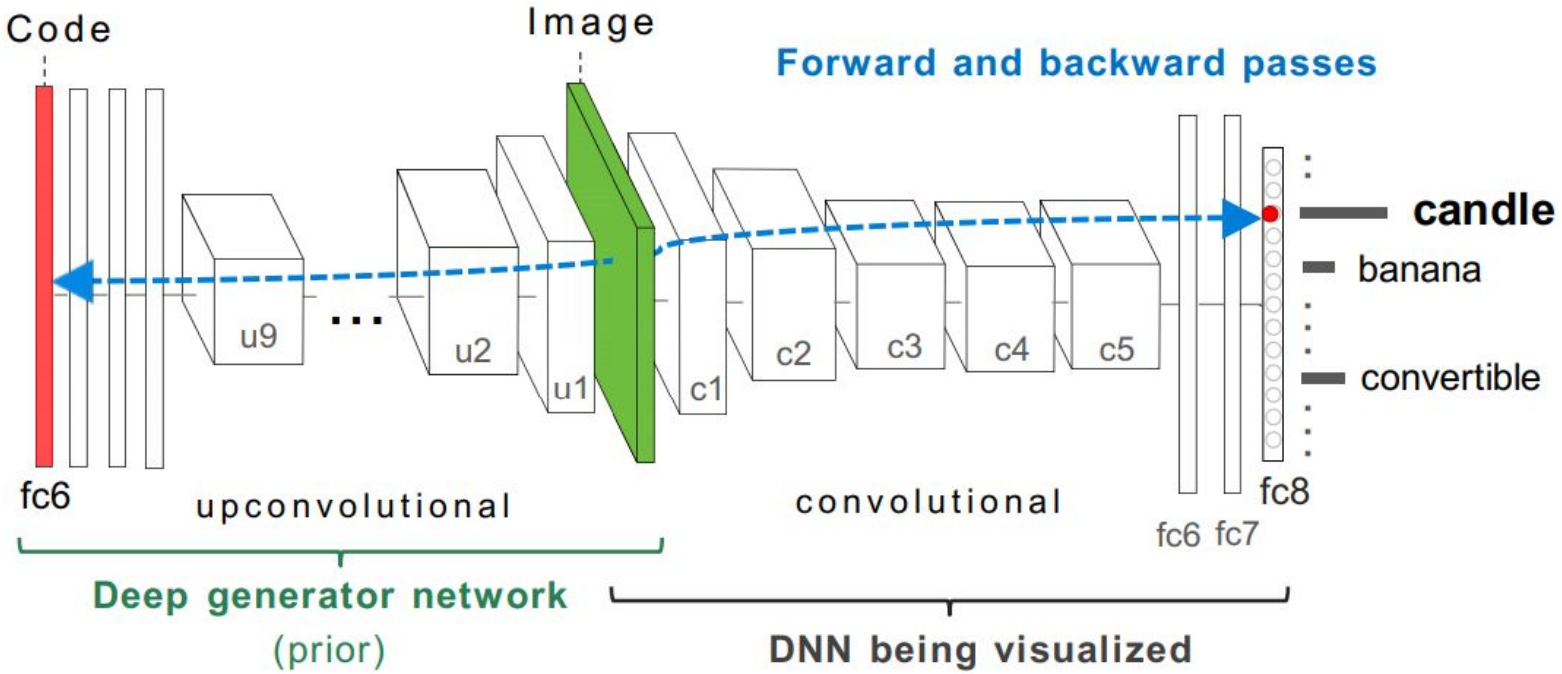
K. Simonyan, A. Vedaldi, A. Ziseramnn, Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, ICLR 2014

Activation Maximization (AM): Regularizations

	Unregularized	Frequency Penalization	Transformation Robustness	Learned Prior	Dataset Examples
 **Erhan, *et al.*, 2009** [3] Introduced core idea. Minimal regularization.	■				
 **Szegedy, *et al.*, 2013** [11] Adversarial examples. Visualizes with dataset examples.	■				■
 **Mahendran & Vedaldi, 2015** [7] Introduces total variation regularizer. Reconstructs input from representation.		■			
 **Nguyen, *et al.*, 2015** [14] Explores counterexamples. Introduces image blurring.		■			
 **Mordvintsev, *et al.*, 2015** [4] Introduced jitter & multi-scale. Explored GMM priors for classes.			■	■	
 **Øygard, *et al.*, 2015** [15] Introduces gradient blurring. (Also uses jitter.)		■	■		
 **Tyka, *et al.*, 2016** [16] Regularizes with bilateral filters. (Also uses jitter.)		■	■		
 **Mordvintsev, *et al.*, 2016** [17] Normalizes gradient frequencies. (Also uses jitter.)		■	■		
 **Nguyen, *et al.*, 2016** [18] Paramaterizes images with GAN generator.				■	
 **Nguyen, *et al.*, 2016** [10] Uses denoising autoencoder prior to make a generative model.				■	

Activation Maximization (AM): Generative prior

Optimizes over the input code (fc6 in the Fig.) instead of the image.



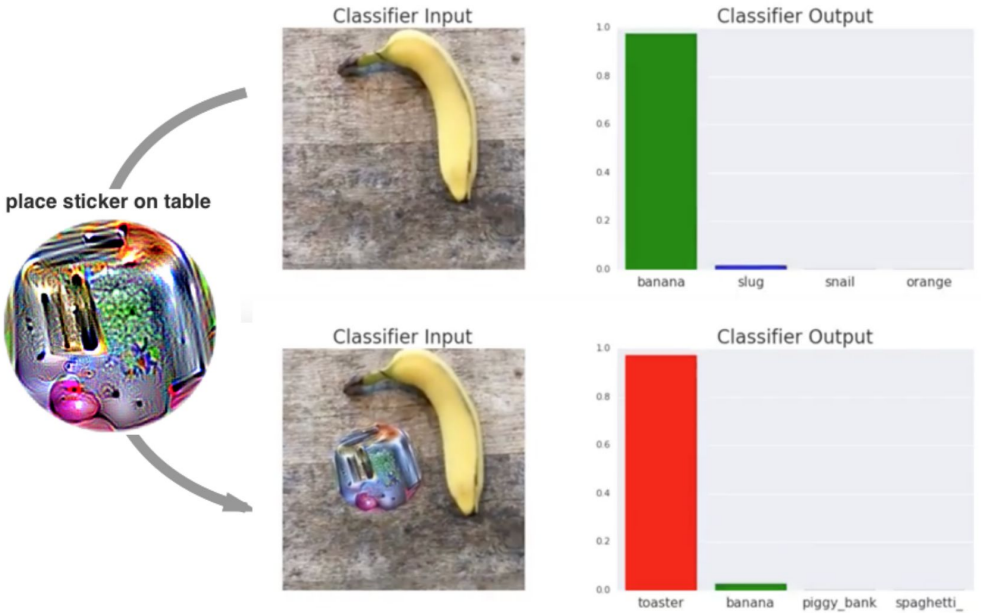
Nguyen, Anh, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. "Synthesizing the preferred inputs for neurons in neural networks via deep generator networks." NIPS 2016.

Activation Maximization (AM): Generative prior

Nguyen, Anh, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. "Synthesizing the preferred inputs for neurons in neural networks via deep generator networks." NIPS 2016.

Adversarial Patches

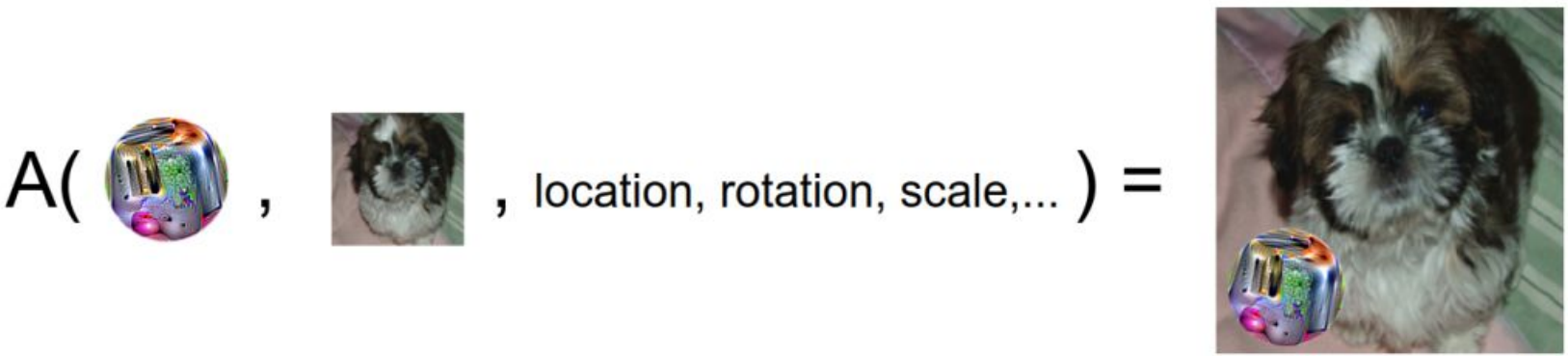
Goal: Generate a patch P that will make a Neural Network N always predict class C for any image containing the patch.



Brown, Tom B., Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. "Adversarial patch." arXiv preprint arXiv:1712.09665 (2017).

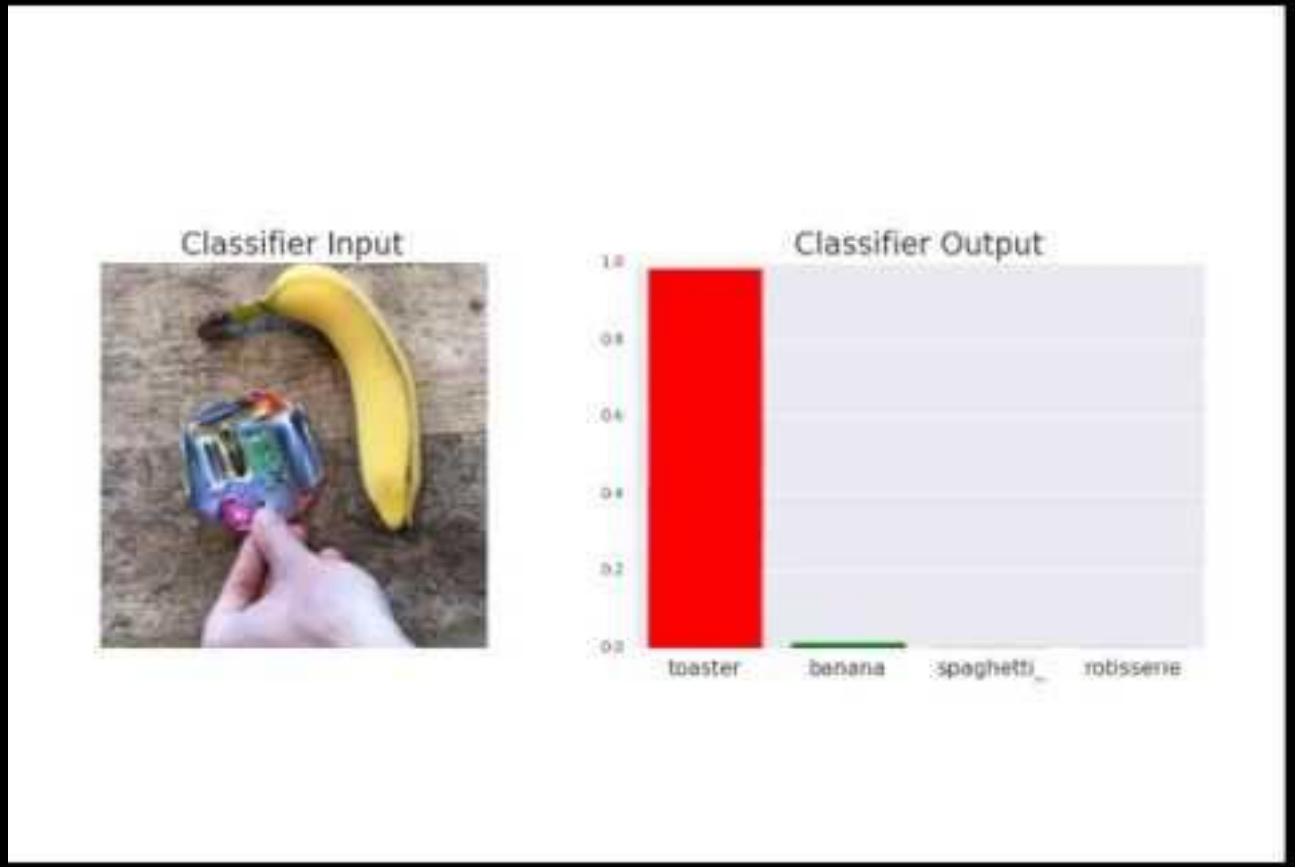
Adversarial Patches

How: Backpropagate the gradient to images where the patch has been inserted.



Brown, Tom B., Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. "Adversarial patch." arXiv preprint arXiv:1712.09665 (2017).

Sachin Joglekar, "Adversarial patches for CNNs explained" (2018)

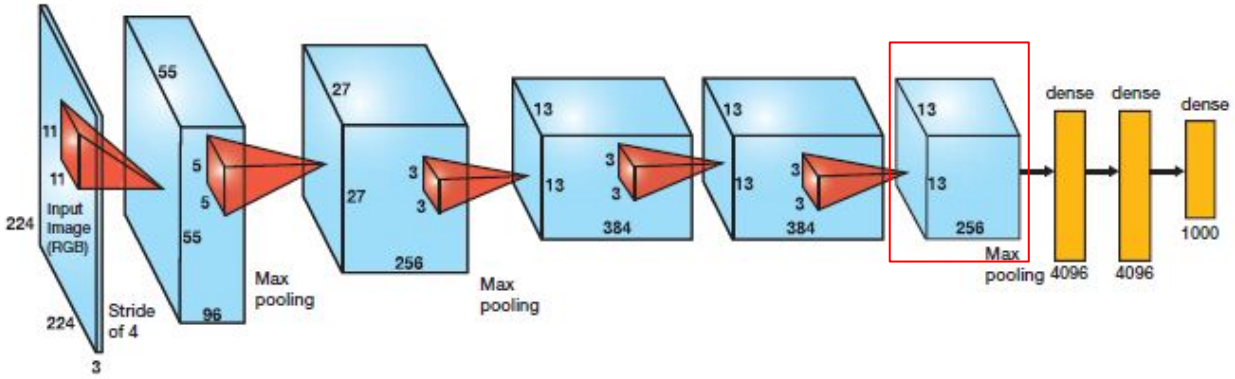


Brown, Tom B., Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. "Adversarial patch." arXiv preprint arXiv:1712.09665 (2017).

Extra: DeepDream

https://github.com/google/deepdream

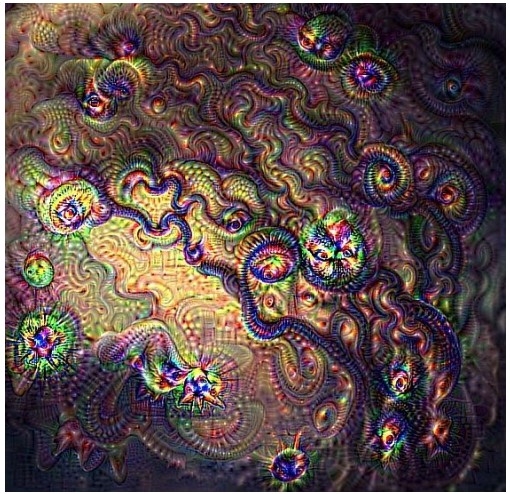
Extra: DeepDream



1. Forward image up to some layer (e.g. conv5)
2. Set the gradients to equal the layer activations
3. Backprop to get gradient on the image
4. Update image (small step in the gradient direction)
5. Repeat

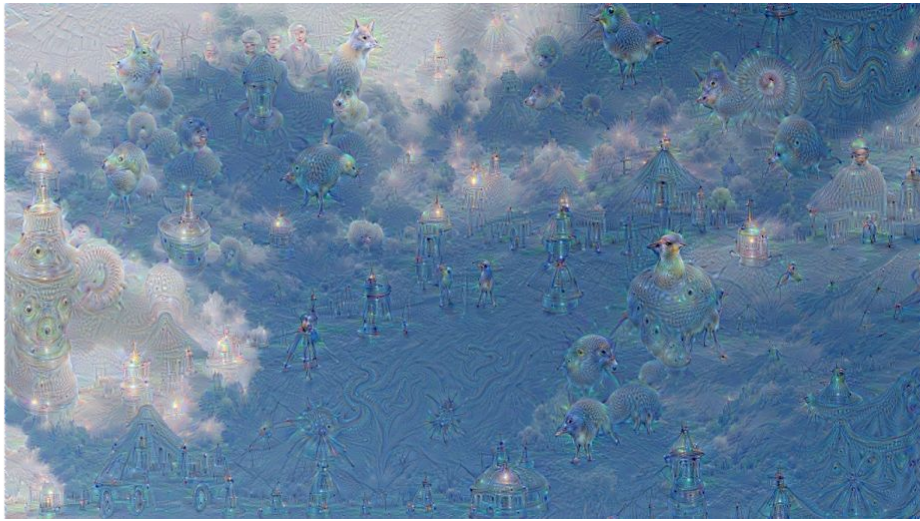
Extra: DeepDream

1. Forward image up to some layer (e.g. conv5)
2. **Set the gradients to equal the layer activations**
3. Backprop to get gradient on the image
4. Update image (small step in the gradient direction)
5. Repeat

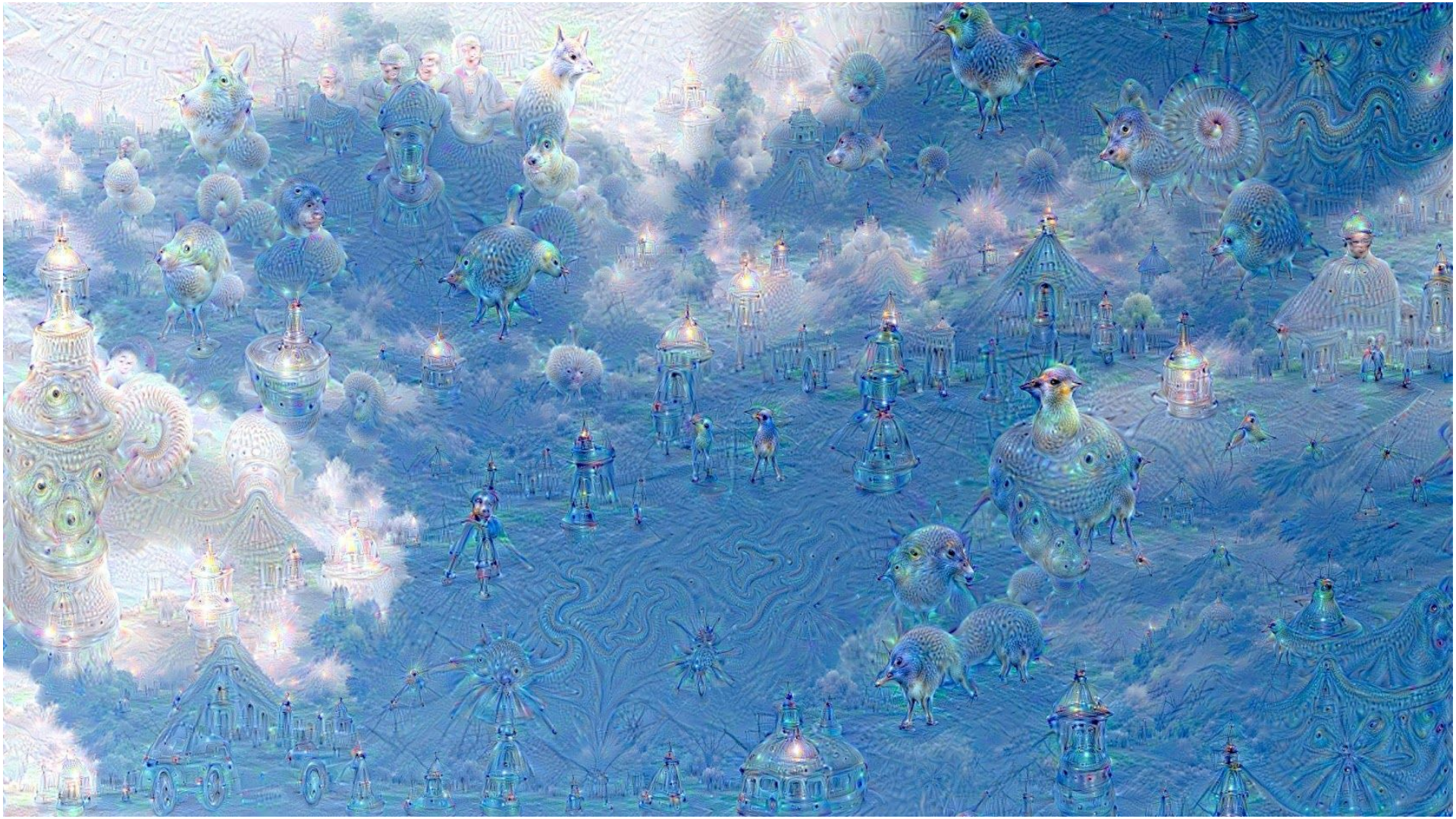


At each iteration, the image is updated to **boost all features** that activated in that layer in the forward pass.

Extra: DeepDream



More examples here



Extra: DeepDream

The network tends to generate animals because it was trained with many pictures of animals.

"Admiral Dog!"

"The Pig-Snail"

"The Camel-Bird"

"The Dog-Fish"

Google Research, “Going deeper into neural networks” - DeepDream (2015)

Extra: DeepDream

If we apply the algorithm iteratively on its own outputs and apply some zooming after each iteration, we get an endless stream of new impressions, exploring the set of things the network knows about.

Google Research, “Going deeper into neural networks” - DeepDream (2015)

Video: Jonah Nordberg. http://johan-nordberg.com/

Overview

- Learned weights
- Activations from data
- Gradient-based
- Activation Maximization (AM)

Questions?