Insight
Centre for Data Analytics

DEEP LEARNING WORKSHOP

Dublin City University
21-22 May 2018

DCU

Day 1 Lecture 5

Visualization

Amaia Salvador
amaia.salvador@upc.edu

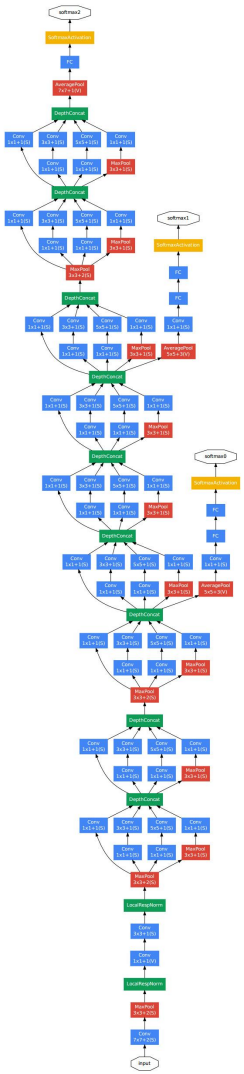
PhD Candidate
Universitat Politècnica de Catalunya

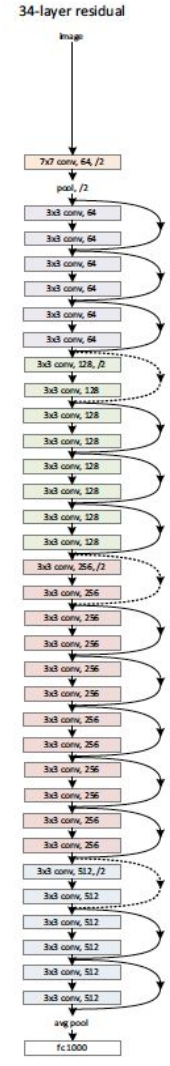
UPC

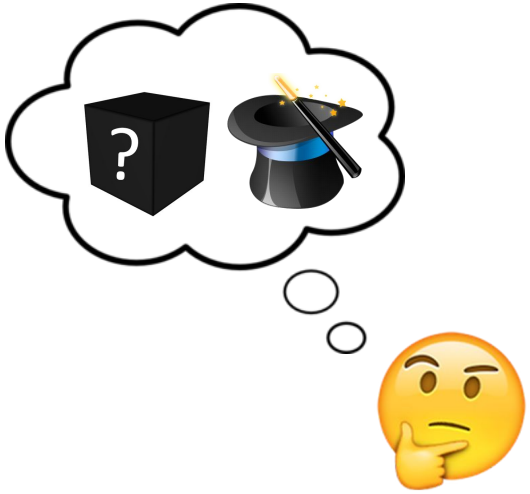
AlexNet

image
conv-64
conv-192
conv-384
conv-256
conv-256
FC-4096
FC-4096
FC-1000

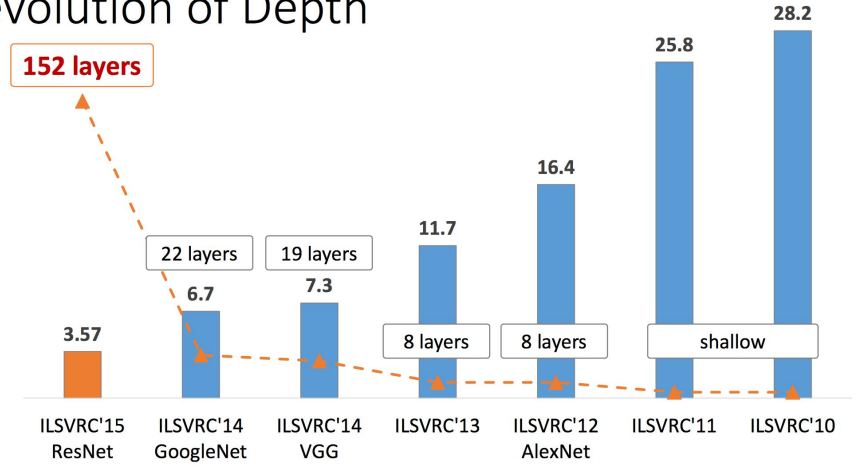
image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
conv-256
conv-256
maxpool
conv-512
conv-512
conv-512
conv-512
maxpool
conv-512
conv-512
conv-512
conv-512
maxpool
FC-4096
FC-4096
FC-1000
softmax







Revolution of Depth



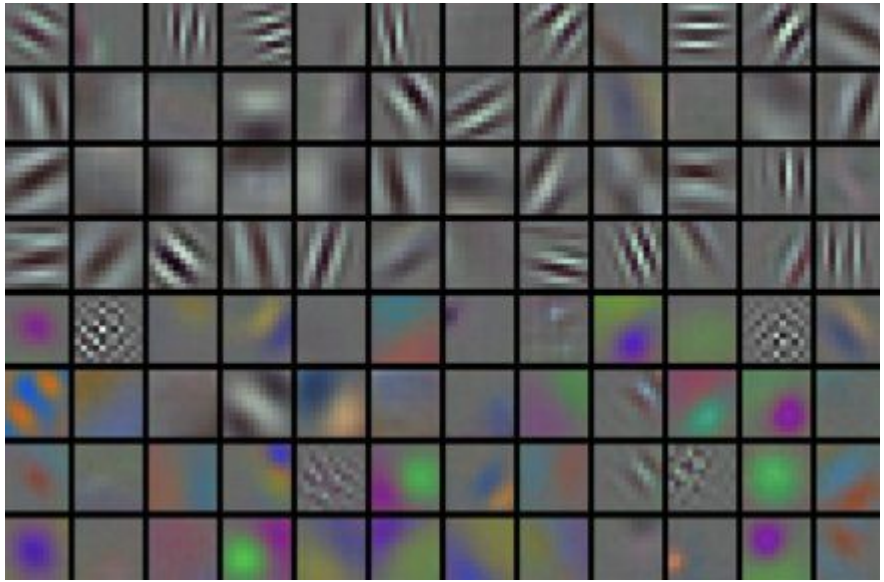
Visualization

- Learned weights
- Activations from data
- Gradient-based
- Optimization-based

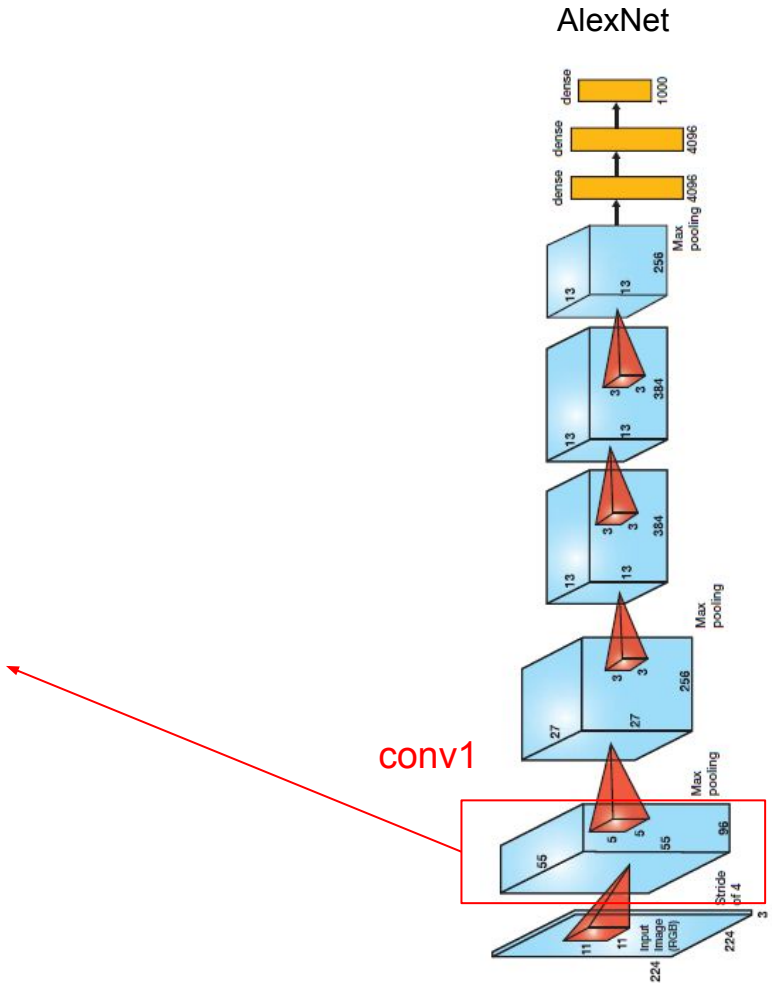
Visualization

- **Learned weights**
- Activations from data
- Gradient-based
- Optimization-based

Visualize Learned Weights



Filters are only “interpretable” on the first layer



Visualize Learned Weights

Weights:

layer 2 weights

Weights:

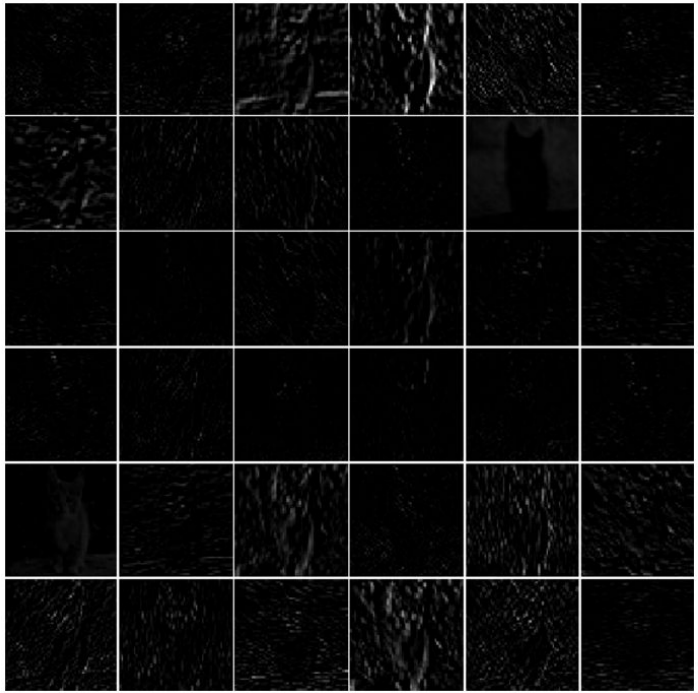
layer 3 weights

Source: ConvnetJS

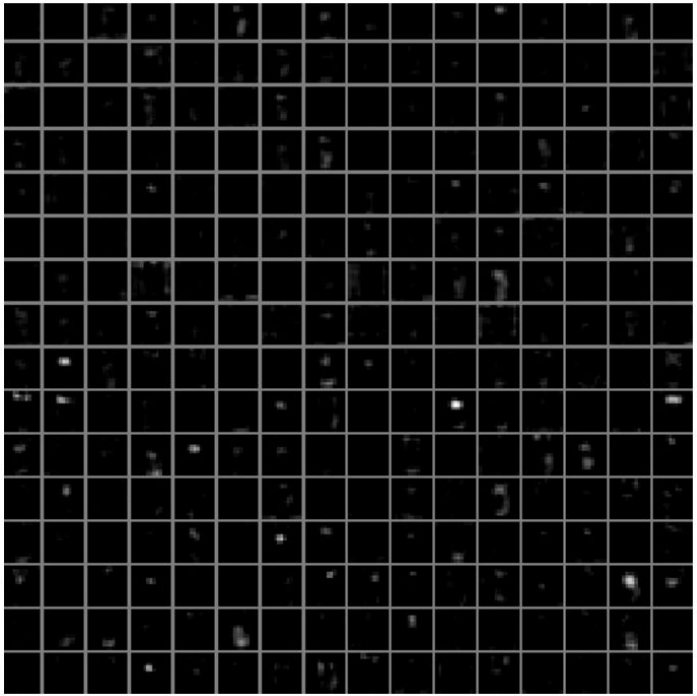
Visualization

- Learned weights
- **Activations from data**
- Gradient-based
- Optimization-based

Activations from data



conv1



conv5

Receptive Field

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

“people”

“text”

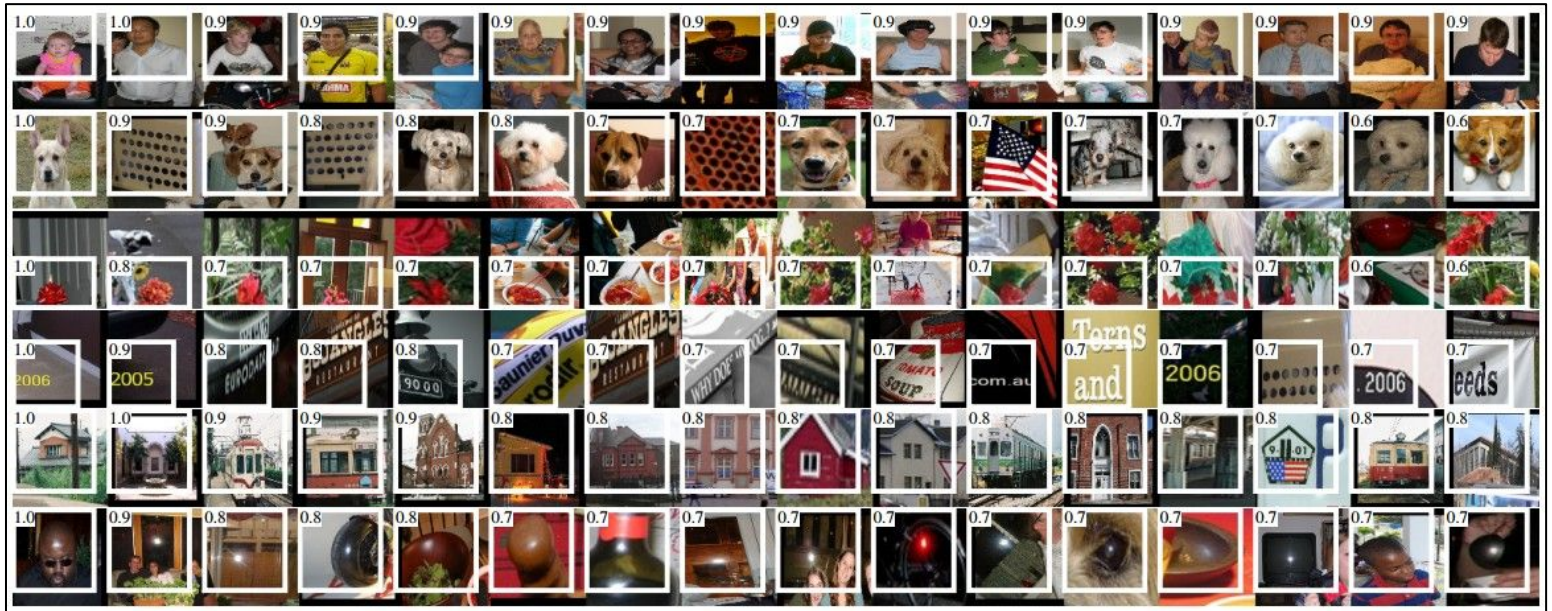
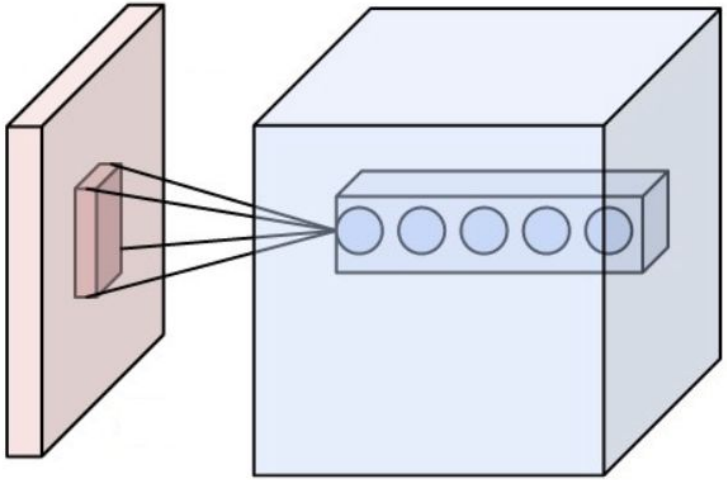


Figure 4: Top regions for six pool$_5$ units. Receptive fields and activation values are drawn in white. Some units are aligned to concepts, such as people (row 1) or text (4). Other units capture texture and material properties, such as dot arrays (2) and specular reflections (6).

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

Reminder: Receptive Field

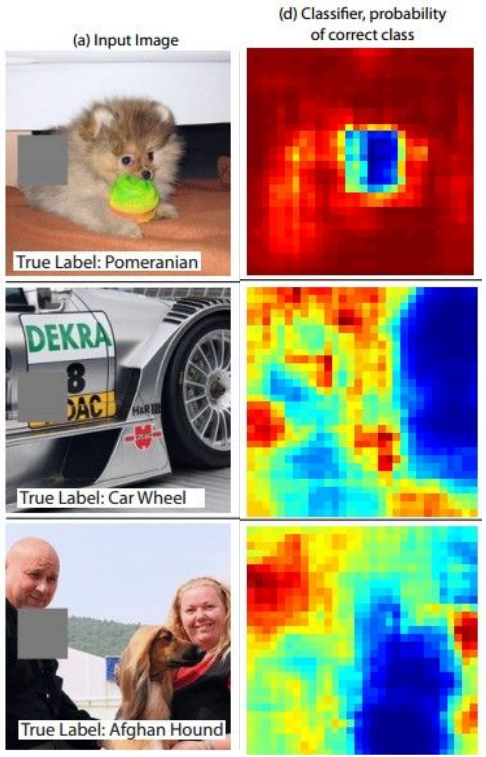


Receptive field: Part of the input that is visible to a neuron. It increases as we stack more convolutional layers (i.e. neurons in deeper layers have larger receptive fields).

Occlusion experiments

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

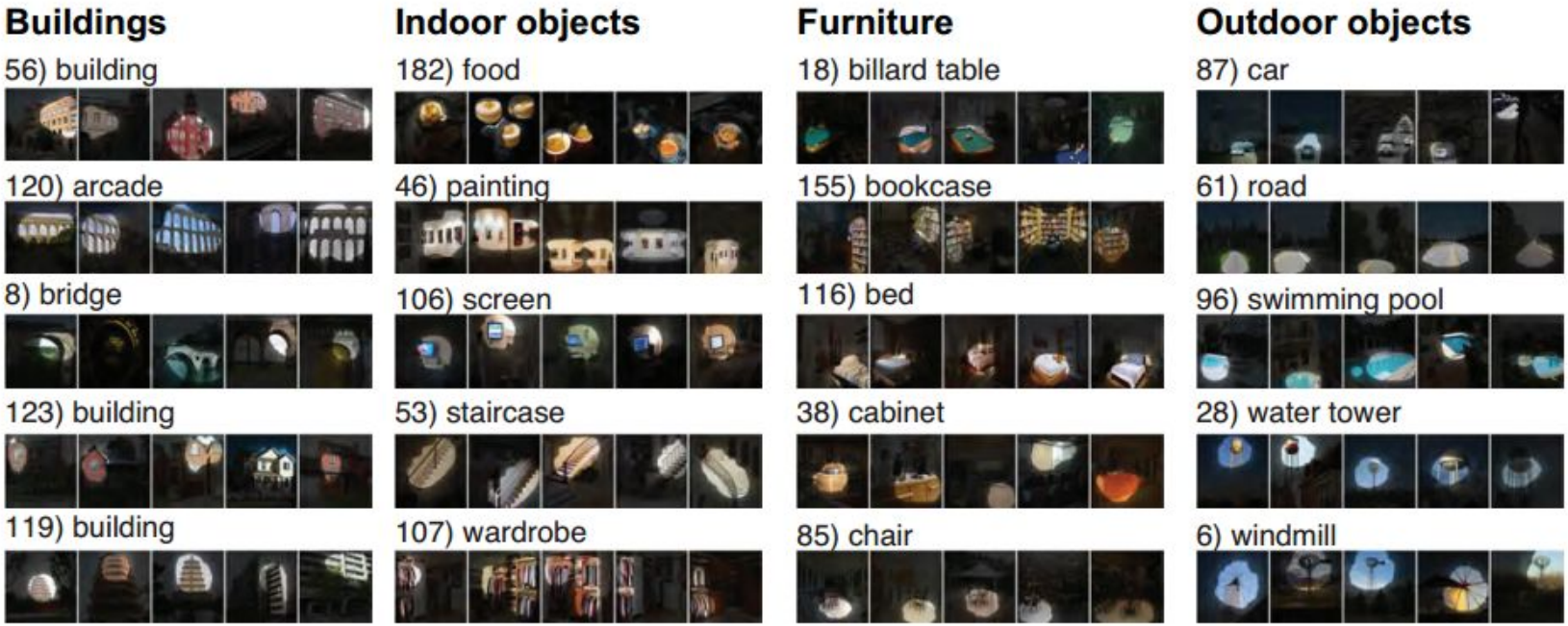
2. Keep track of the probability of the correct class w.r.t. the position of the occluder



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

Occlusion experiments

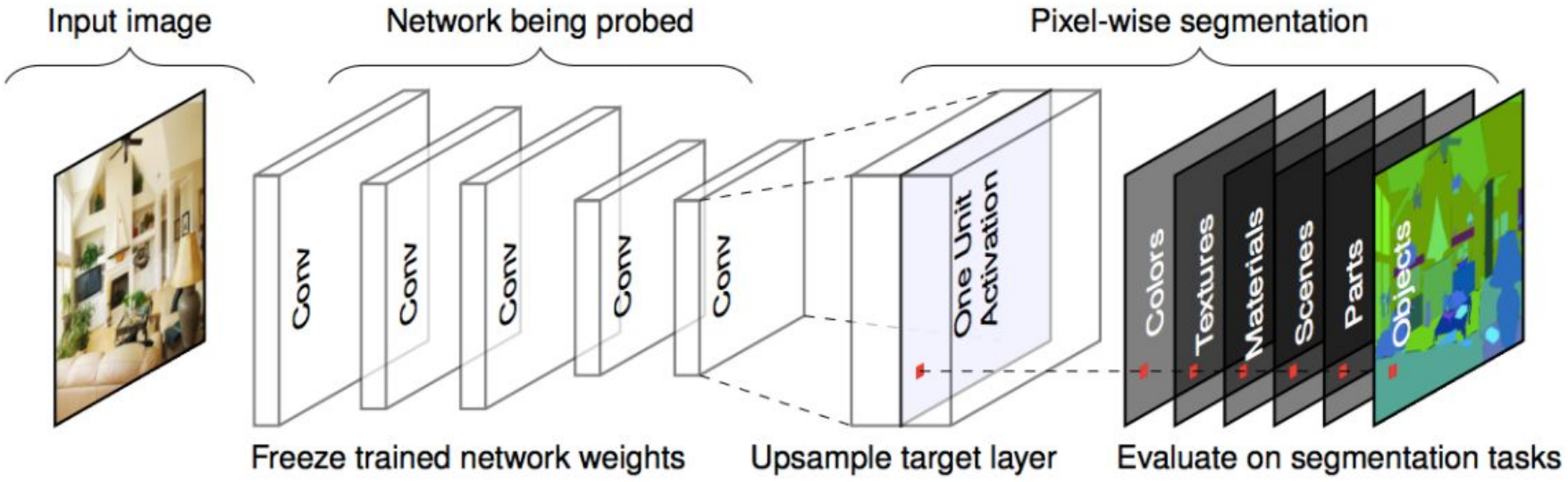
Can be applied to any arbitrary layer. “Manual” label assignment (AMT)



Zhou et al. Object detectors emerge in deep scene CNNs. ICLR 2015

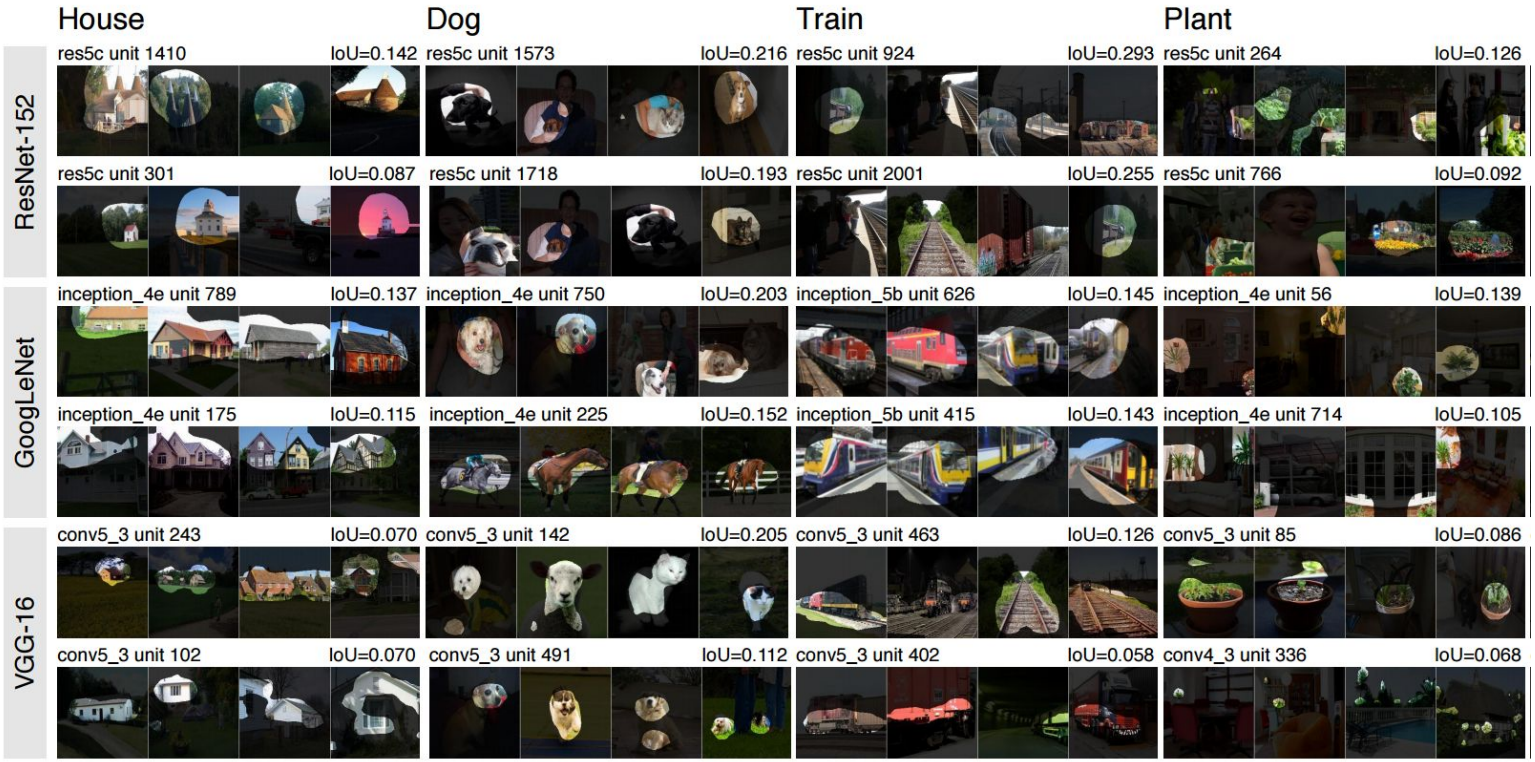
Network Dissection

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

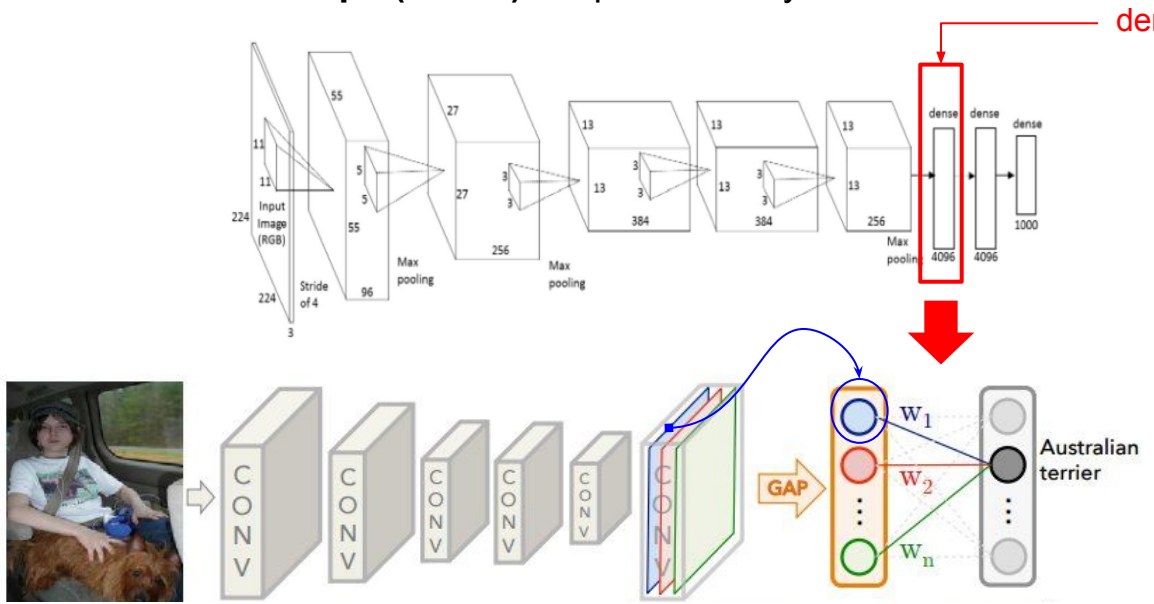
Network Dissection



Bau, Zhou et al. [Network Dissection: Quantifying Interpretability of Deep Visual Representations.](#) CVPR 2017

Class Activation Maps

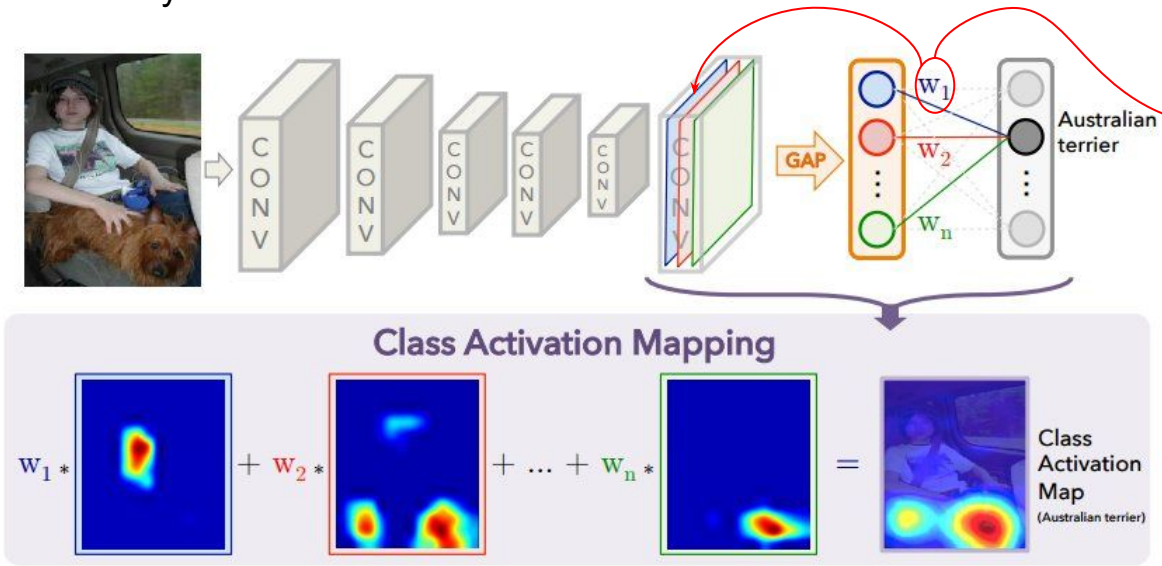
Class Activation Maps (CAMs): Replace FC layer after last conv with Global Average Pooling (GAP).



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

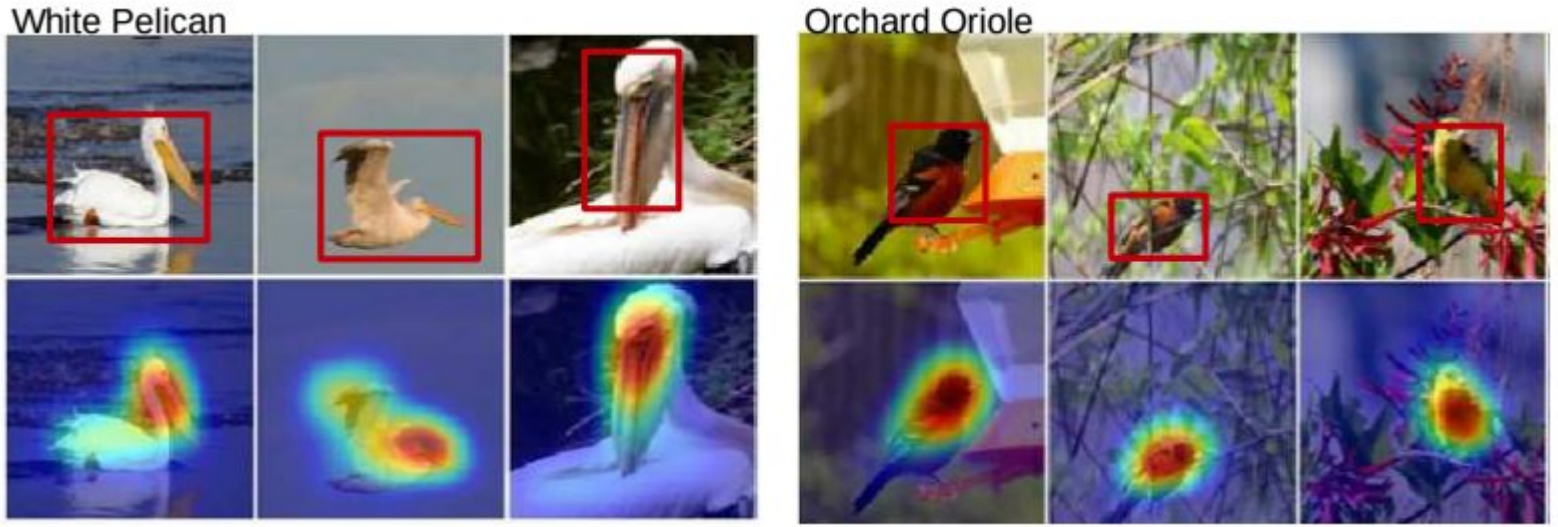


contribution of the *blue* channel in the conv layer to the prediction of "australian terrier"

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



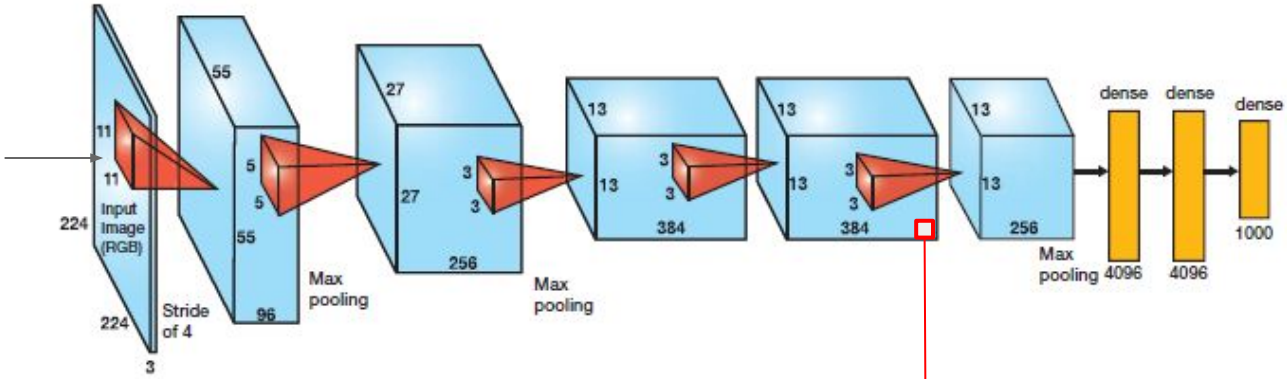
Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. "Learning deep features for discriminative localization." CVPR 2016

Visualization

- Learned weights
- Activations from data
- **Gradient-based**
- Optimization-based

Gradient-based approach

Visualize the part of an image that mostly activates a neuron



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)

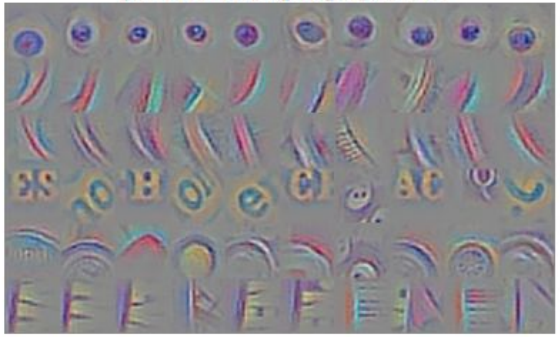
Gradient-based approach

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)

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

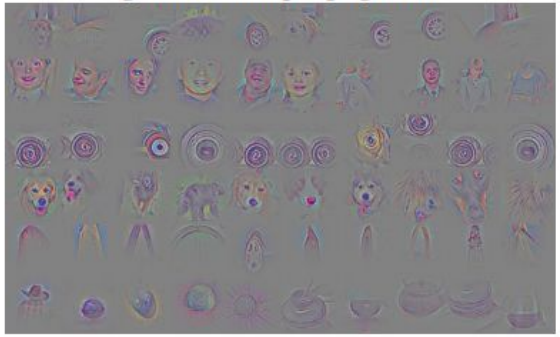
Gradient-based approach

guided backpropagation



corresponding image crops

guided backpropagation



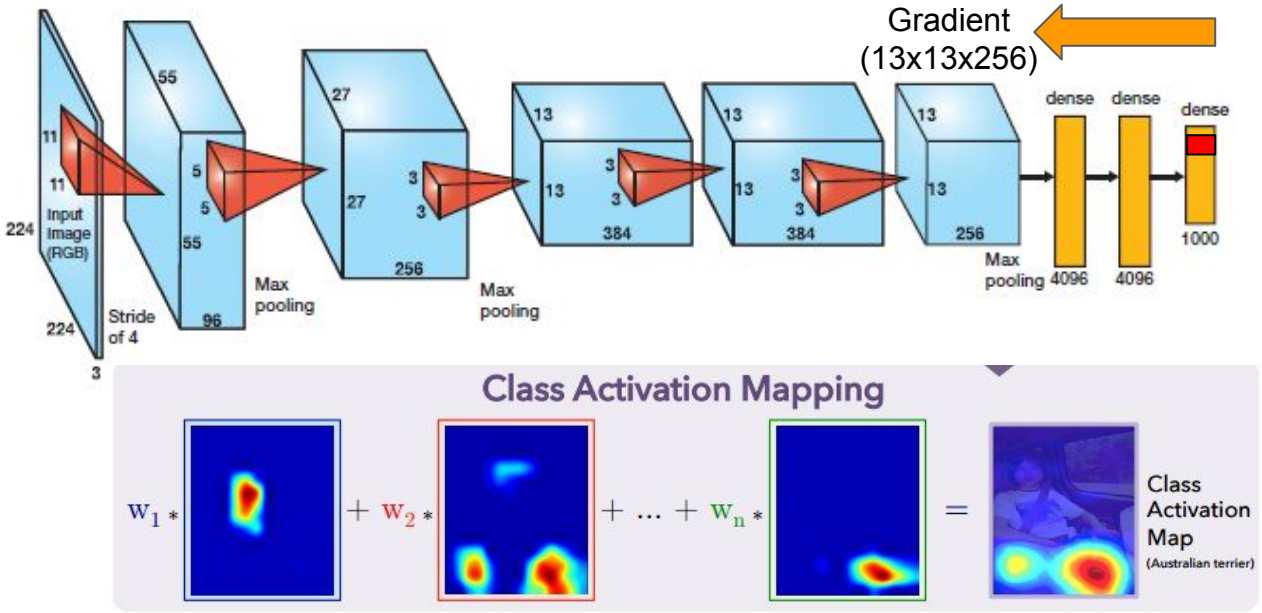
corresponding image crops

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

Grad-CAM

Same idea as CAM, but:

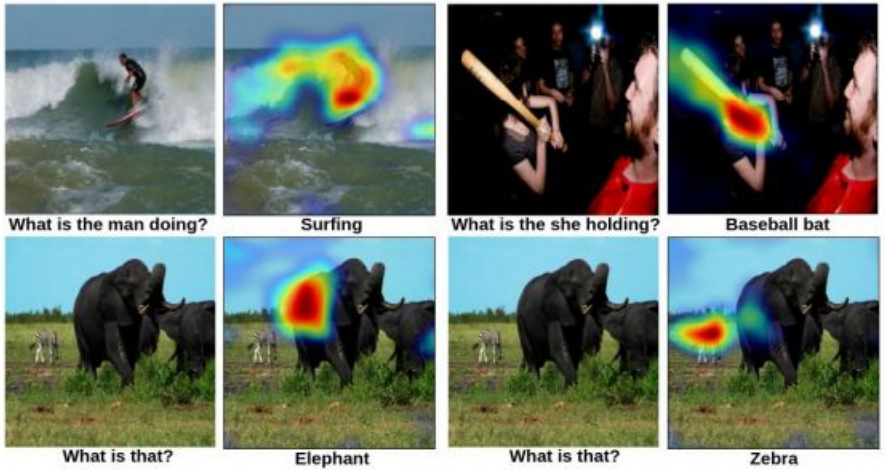
- No modifications required to the network
- Weight feature map by the gradient of the class wrt each channel

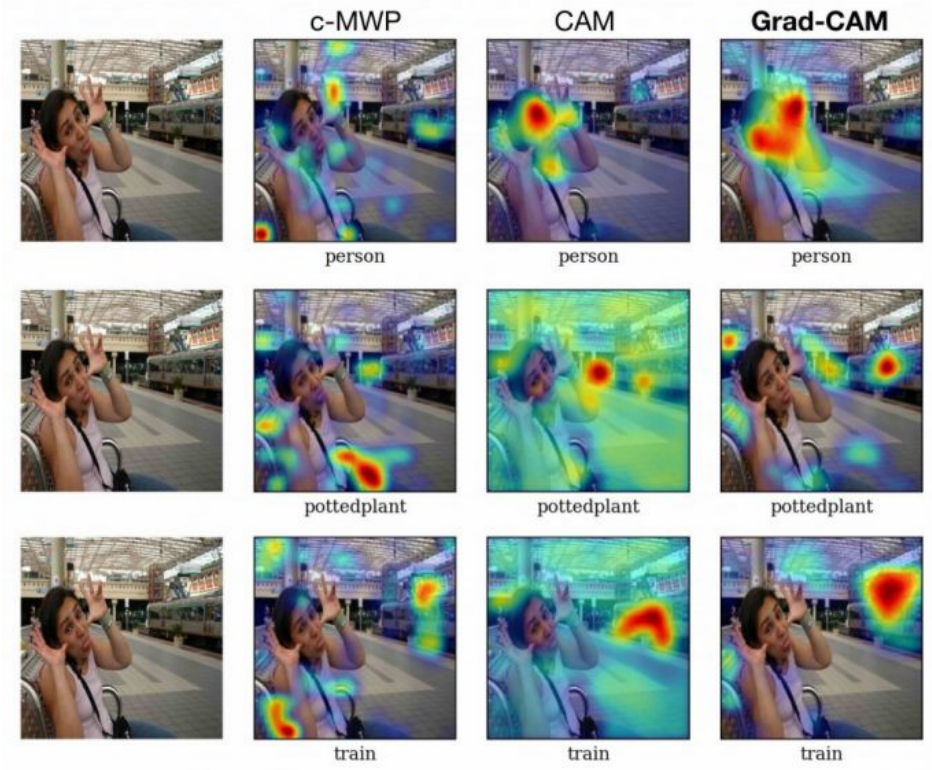


Feature maps weighted by the mean gradient

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

Grad-CAM





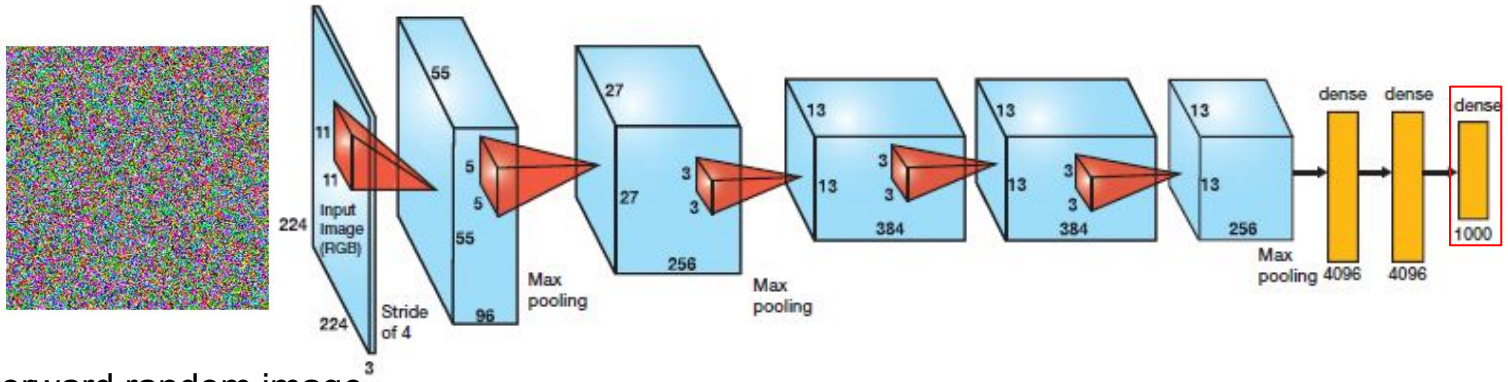
Ramprasaath et al. Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization. ICCV 2017

Visualization

- Learned weights
- Activations from data
- Gradient-based
- **Optimization-based**

Optimization approach

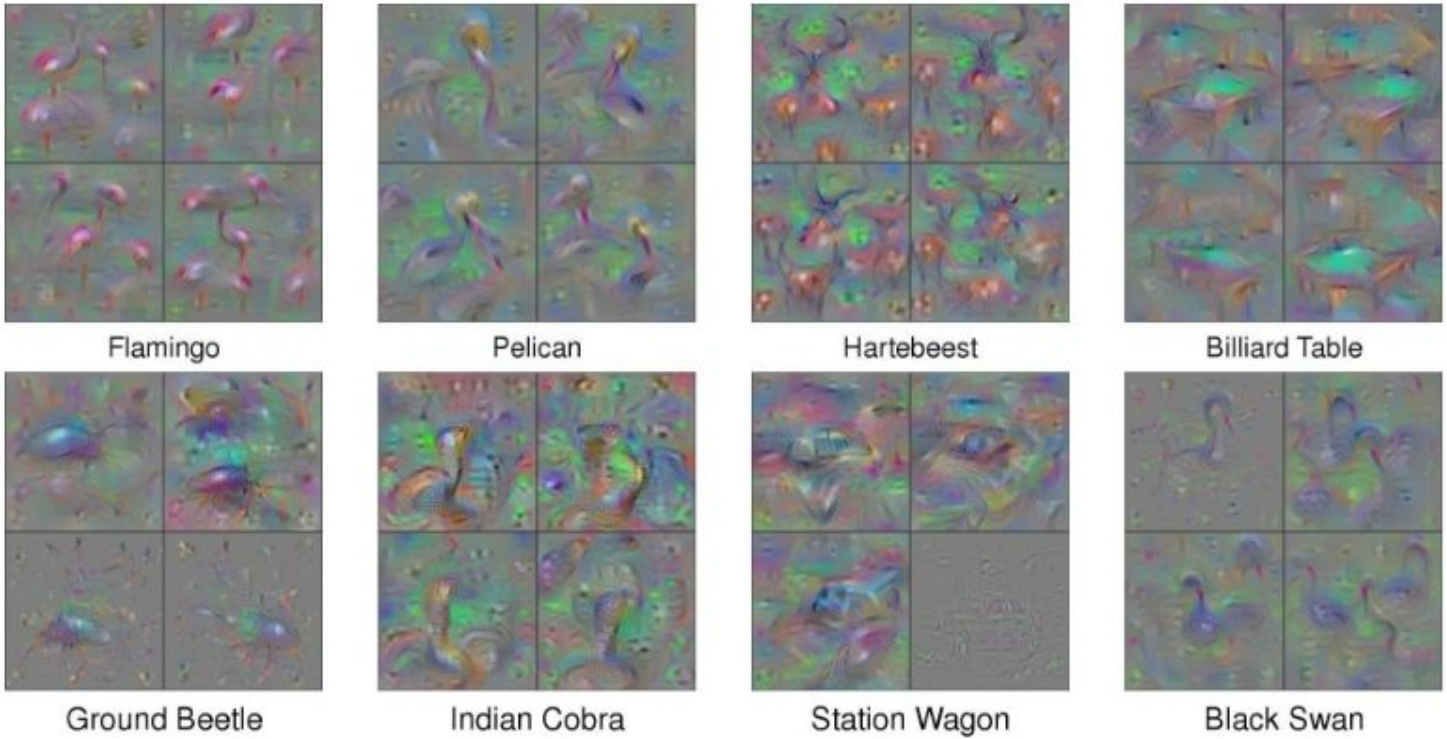
Obtain the image that maximizes a class score (or a neuron activation)



1. Forward random image
2. Set the gradient of the scores 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

Simonyan et al. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, 2014

Optimization approach

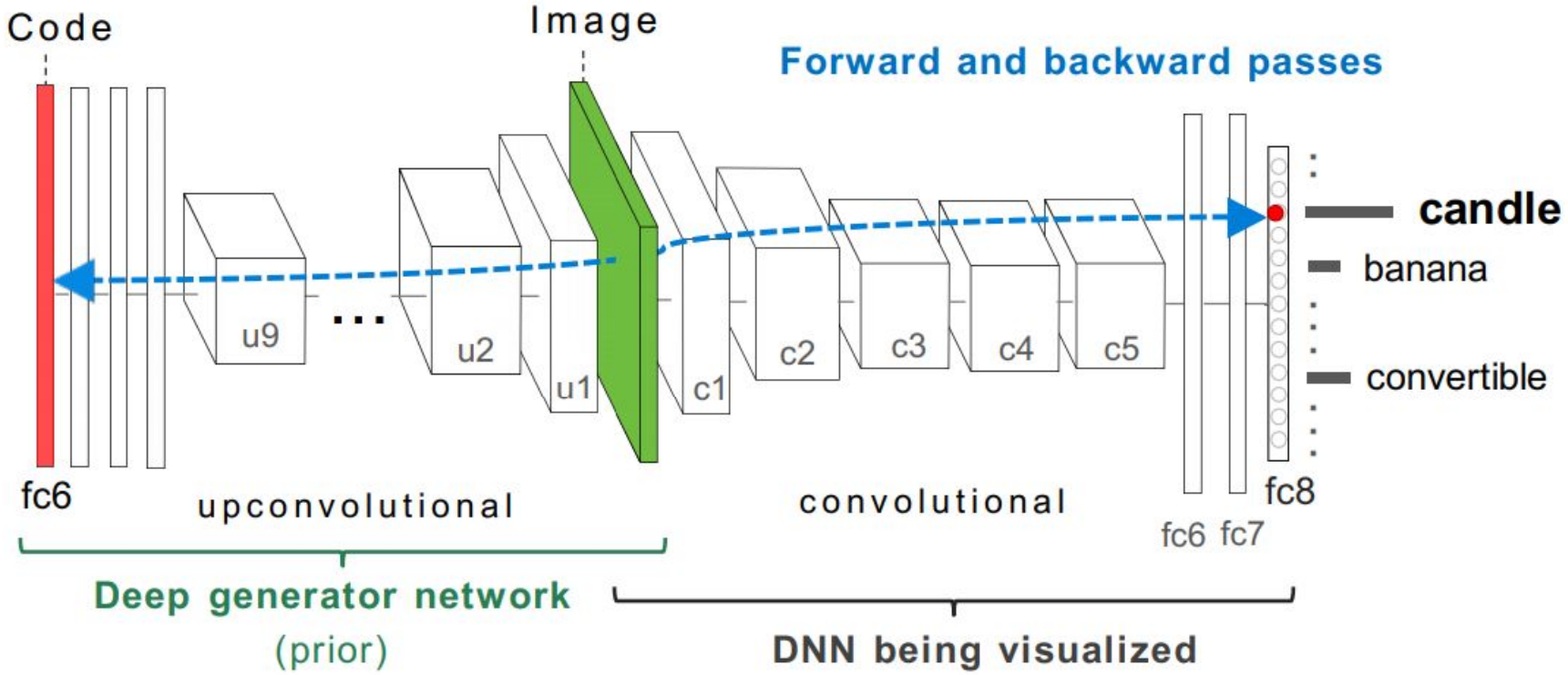


Simonyan et al. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, 2014

Regularization tricks

	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.				■	

Regularization tricks: Generative prior



Nguyen et al. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. NIPS 2016

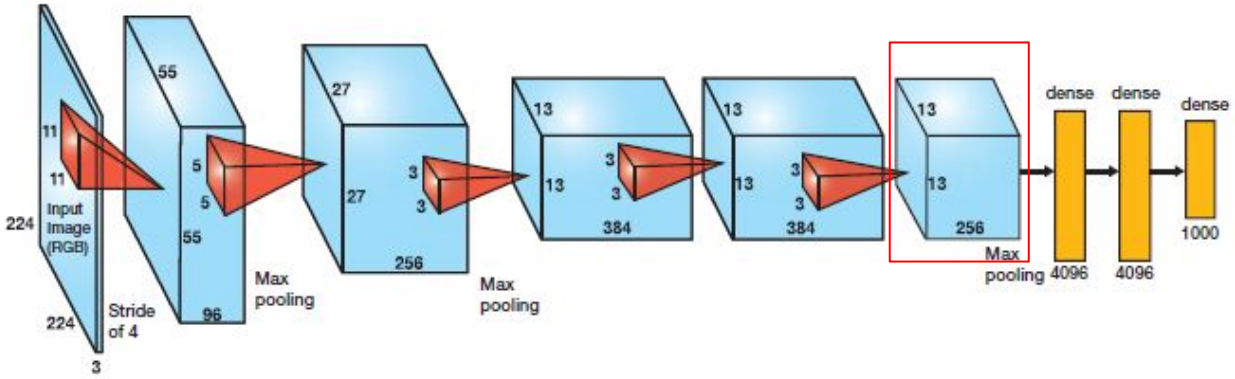
Regularization tricks

Nguyen et al. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. NIPS 2016

Extra: DeepDream

https://github.com/google/deepdream

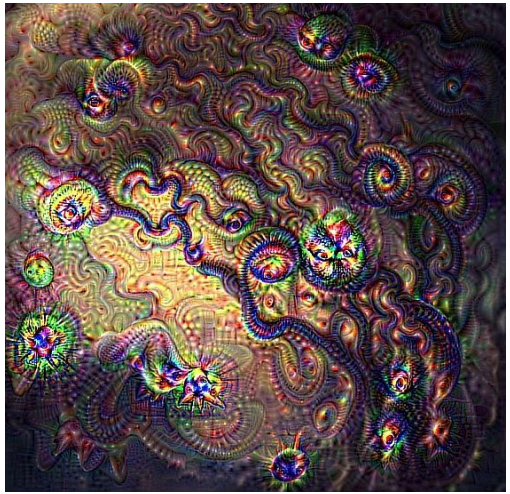
Extra: DeepDream



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

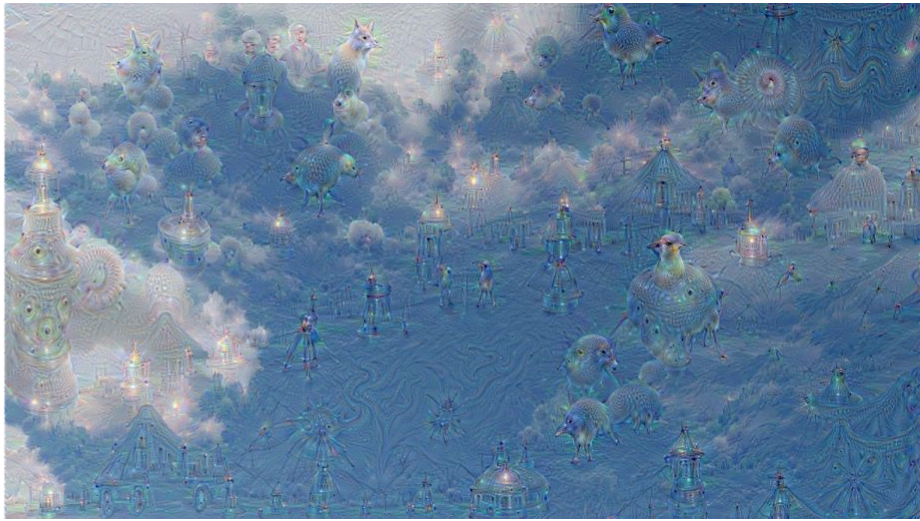
Extra: DeepDream

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

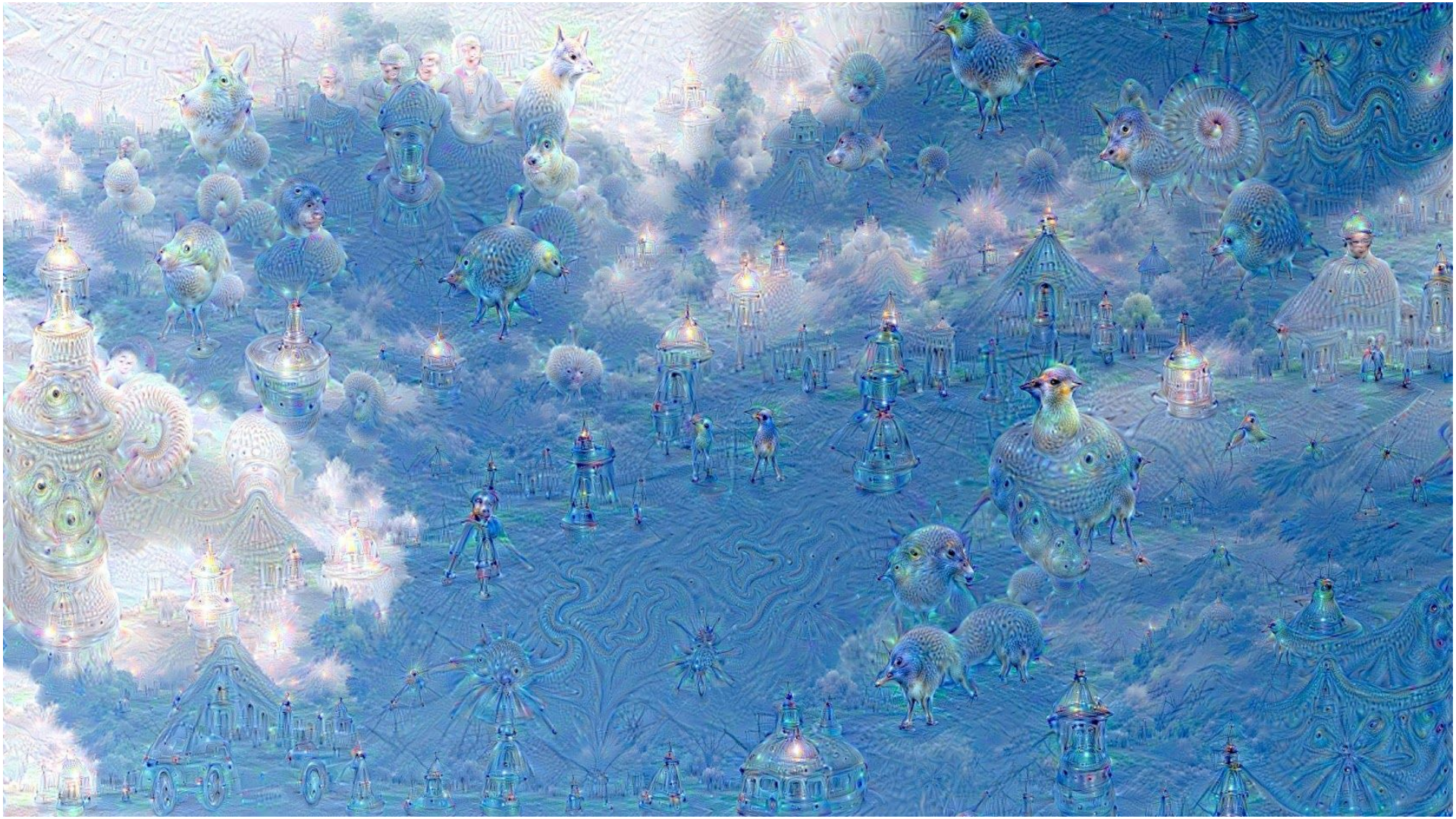


At each iteration, the image is updated to **boost all features** that activated in that layer in the forward pass.

Extra: DeepDream



More examples here



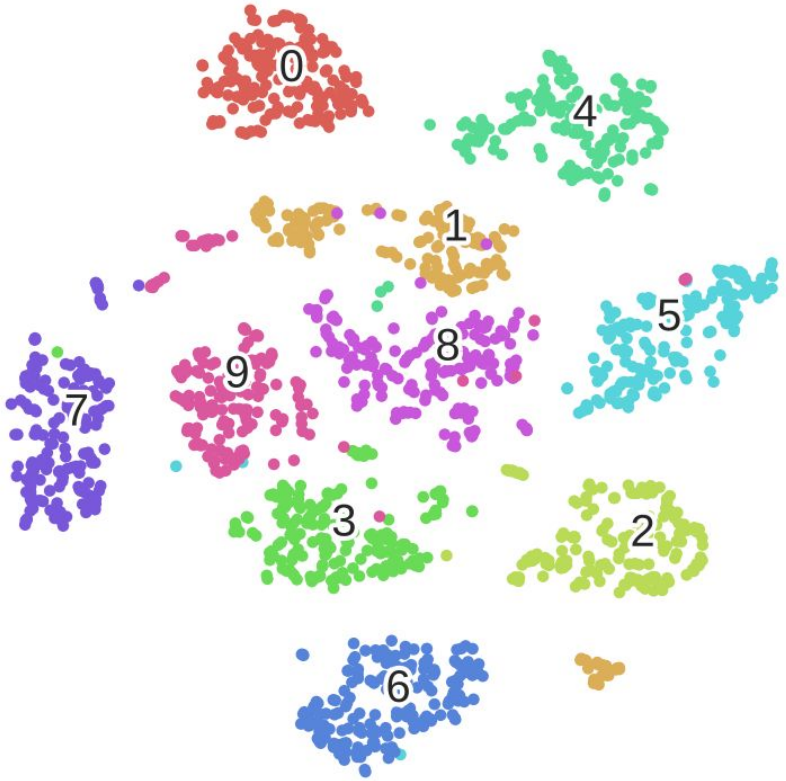
Visualization

- Learned weights
- Activations from data
- Gradient-based
- Optimization-based

Questions?

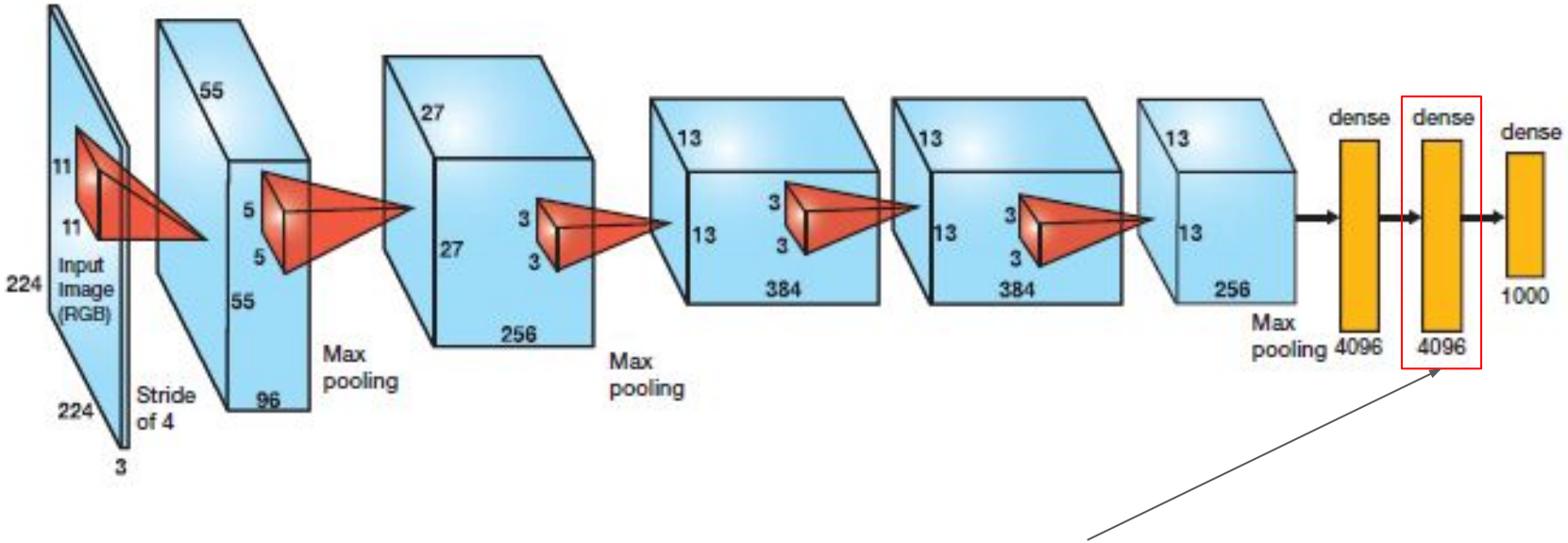
t-SNE

Embed high dimensional data points (i.e. feature codes) so that pairwise distances are preserved in local neighborhoods.



Maaten & Hinton. Visualizing High-Dimensional Data using t-SNE. Journal of Machine Learning Research (2008).

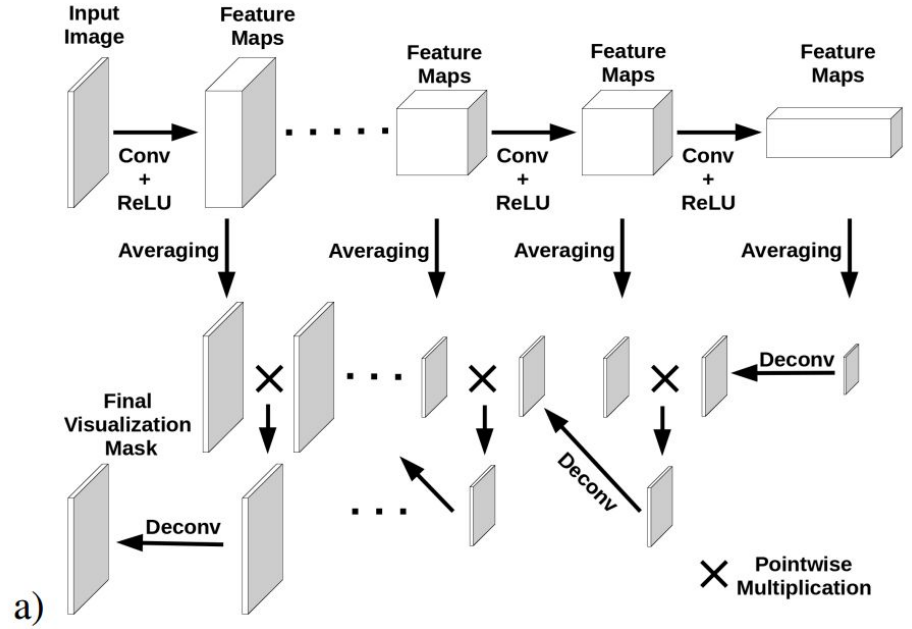
t-SNE



Can be used with features from layer before classification

Optimization approach

Value-based backpropagation

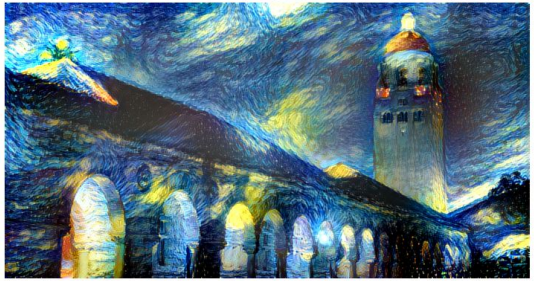


Bojarski et al. VisualBackProp: efficient visualization of CNNs. arXiv May 2017

Neural Style

Style image

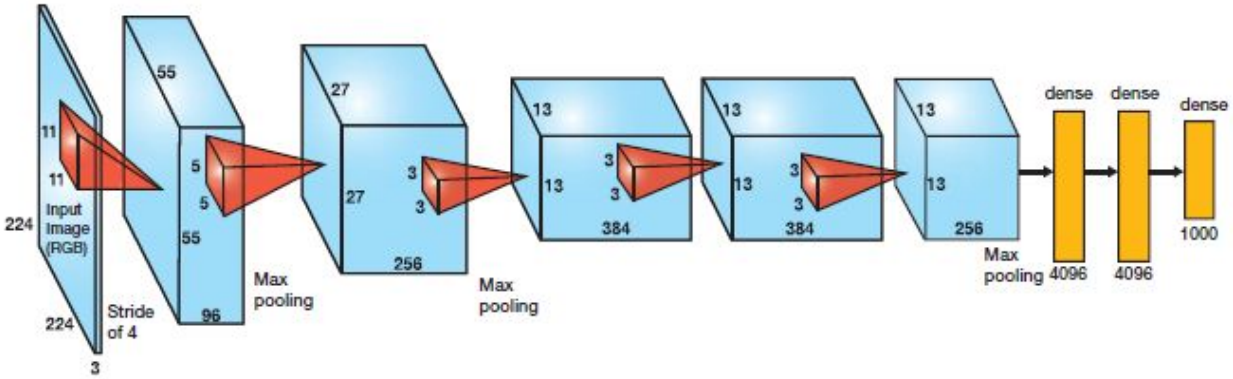
Content image



Result

Gatys et al. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016

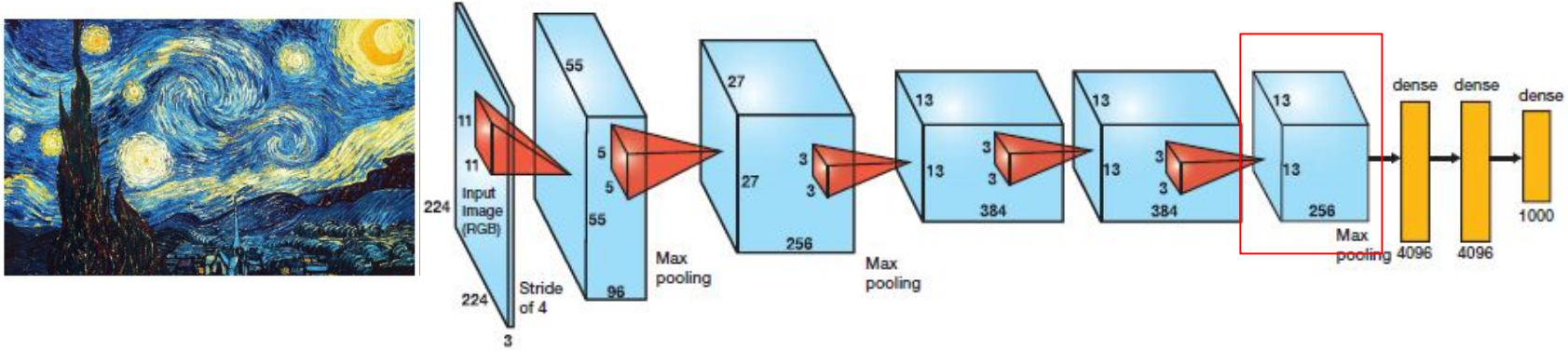
Neural Style



Extract raw activations in all layers. These activations will represent the contents of the image.

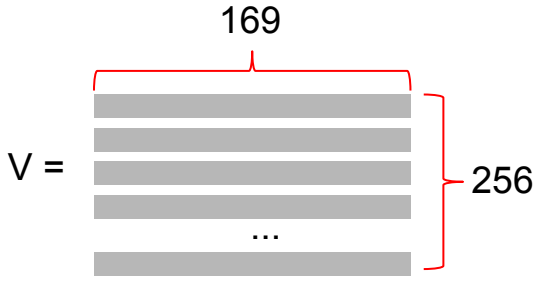
Gatys et al. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016

Neural Style



- Activations are also extracted from the style image for all layers.
- Instead of the raw activations, gram matrices (G) are computed at each layer to represent the style.

E.g. at conv5 [13x13x256], reshape to:

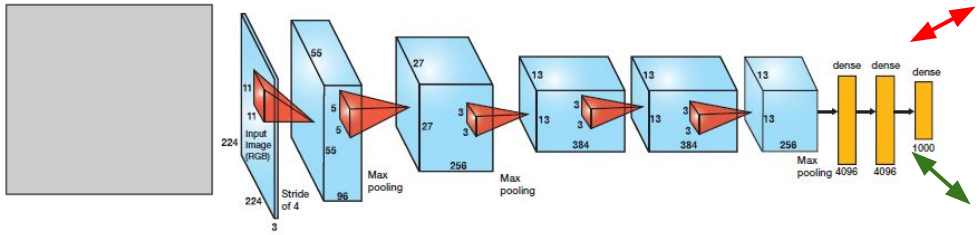


$G = V^T V$

The Gram matrix G gives the correlations between filter responses.

Neural Style

match content



match style

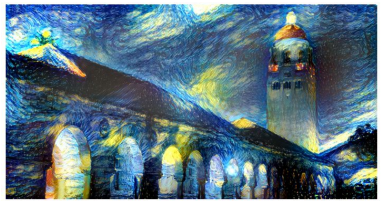
$$\mathcal{L}_{total}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{content}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{style}(\vec{a}, \vec{x})$$

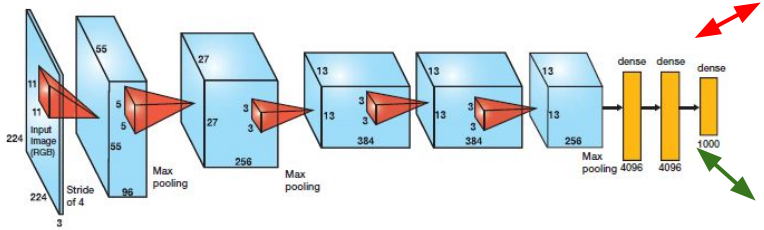
Match activations from content image

Match gram matrices from style image

Gatys et al. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016

Neural Style





match content

match style

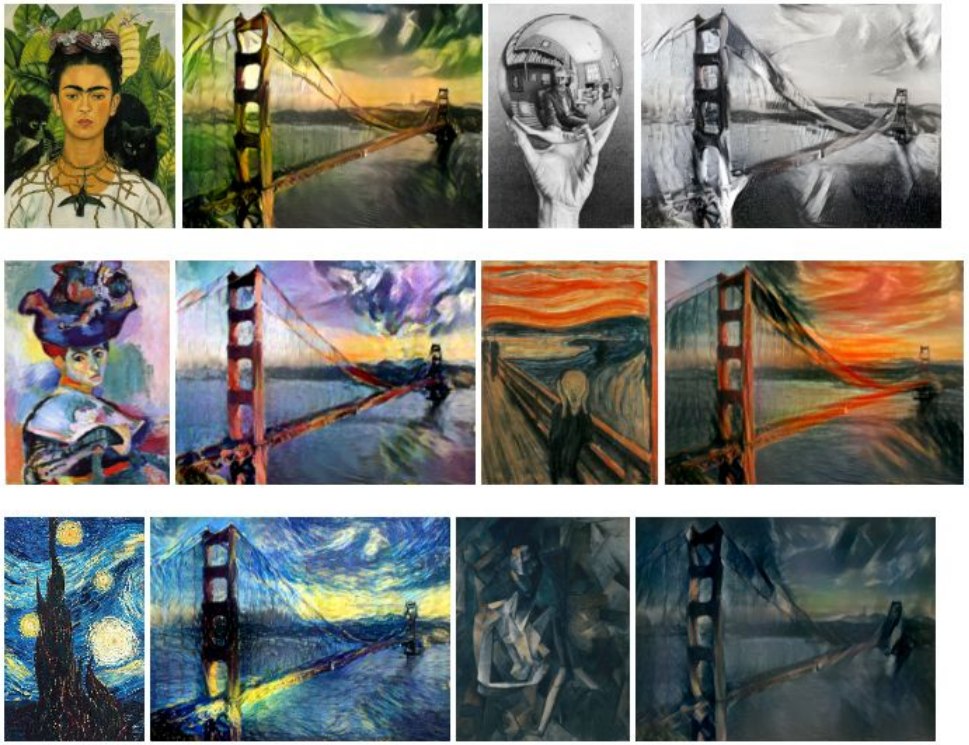
$$\mathcal{L}_{total}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{content}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{style}(\vec{a}, \vec{x})$$

Match activations from content image

Match gram matrices from style image

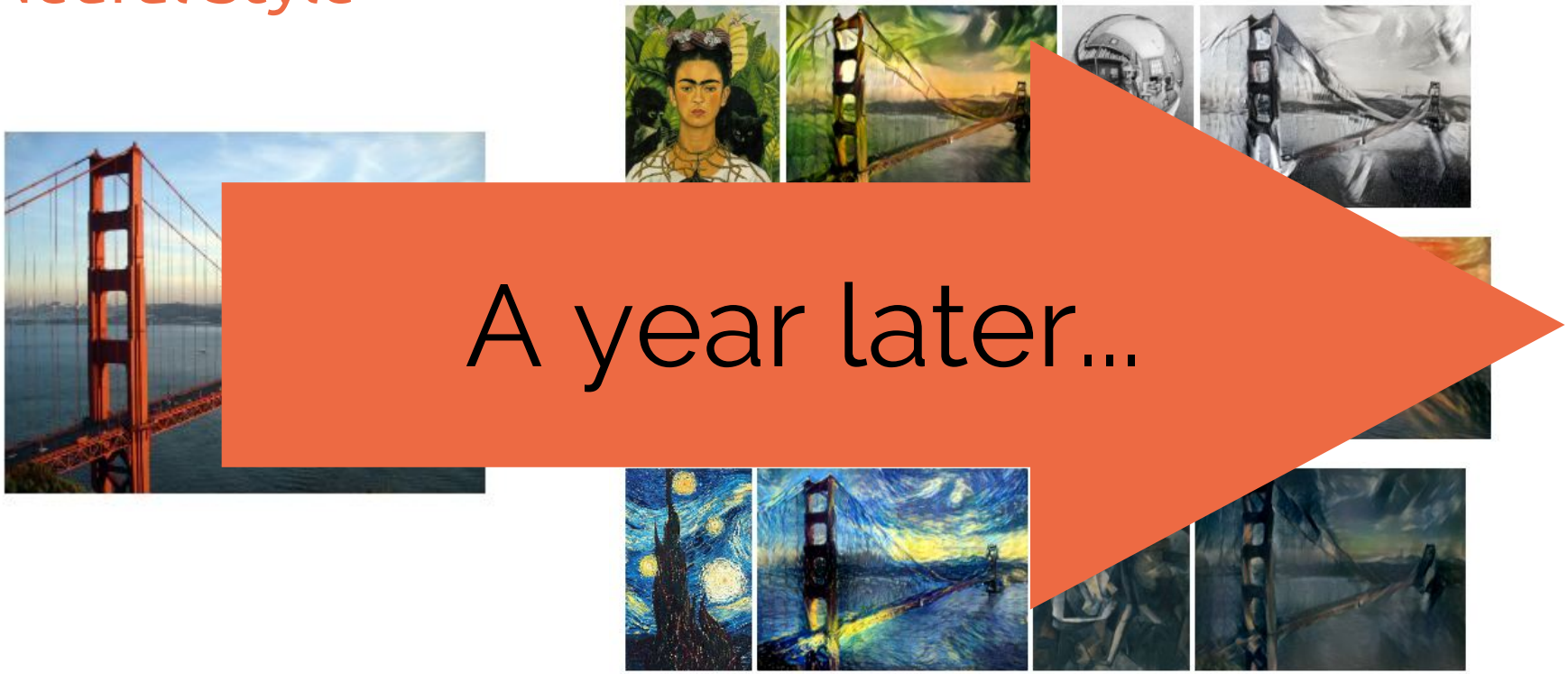
Gatys et al. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016

Neural Style



Gatys et al. [Image Style Transfer Using Convolutional Neural Networks](). CVPR 2016

Neural Style



Gatys et al. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016






Content Image

Style Image

Result

Luan et al. Deep Photo Style Transfer. arXiv Apr 2017






Content Image

Style Image

Result

Luan et al. Deep Photo Style Transfer. arXiv Apr 2017