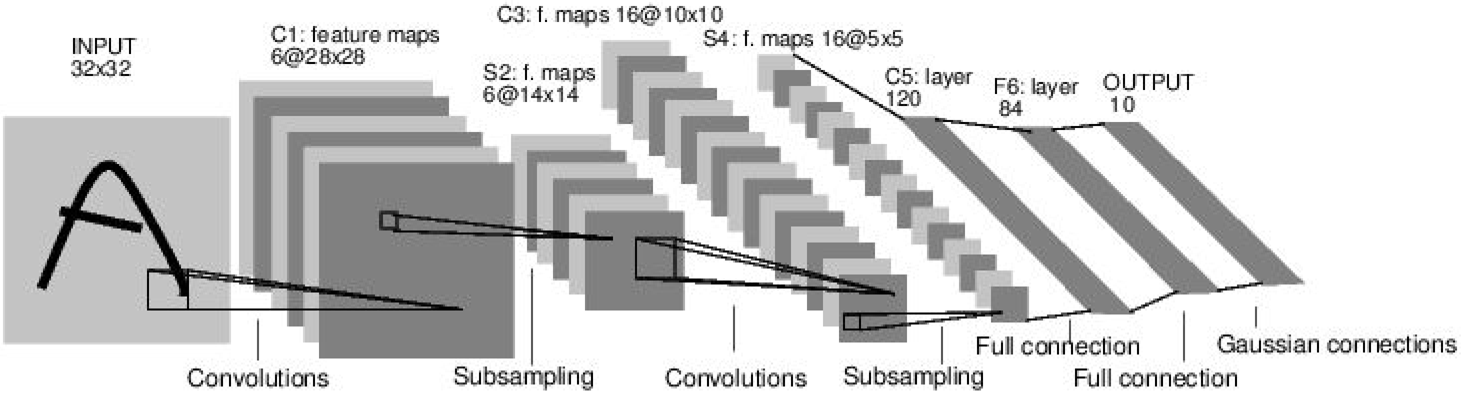
Day 2 Lecture 3

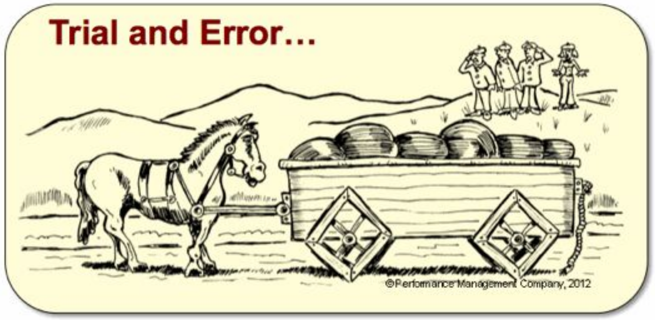
Visualization

Visualization



Understand what ConvNets learn

Visualization



The development of better convnets is reduced to trial-and-error.

Visualization can help in proposing better architectures.

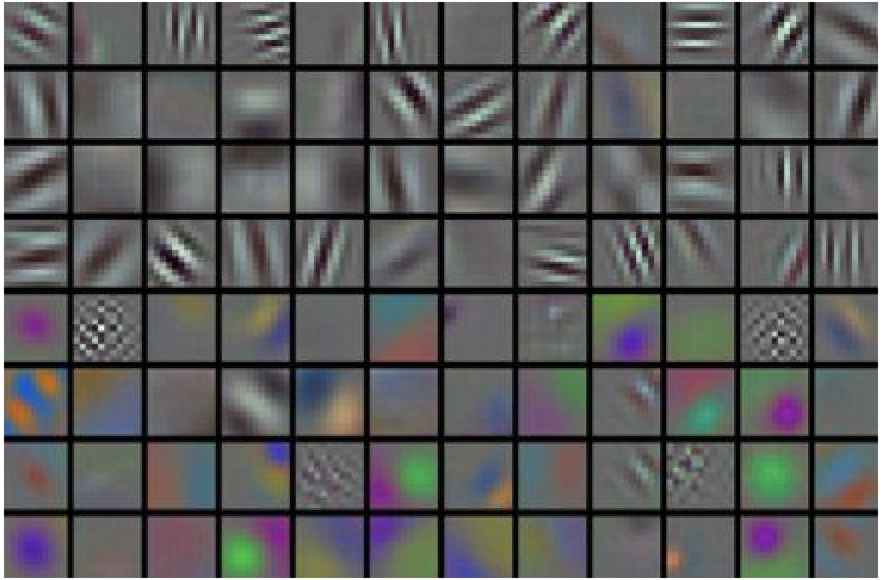
Visualization

- Learned weights
- Activations from data
- Representation space
- Deconvolution-based
- Optimization-based
- DeepDream
- Neural Style

Visualization

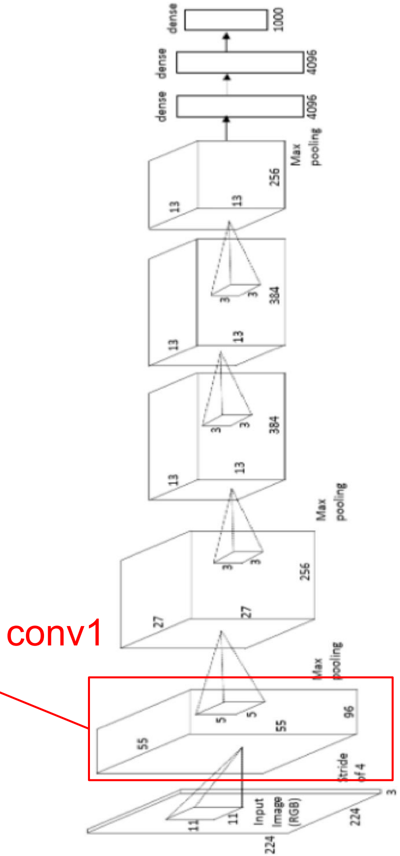
- **Learned weights**
- Activations from data
- Representation space
- Deconvolution-based
- Optimization-based
- DeepDream
- Neural Style

Visualize Learned Weights

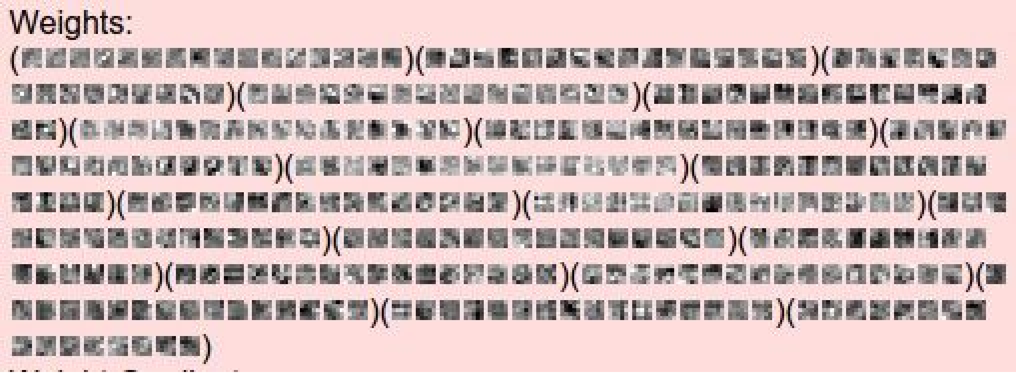


Filters are only interpretable on the first layer

AlexNet



Visualize Learned Weights



layer 2 weights

layer 3 weights

Source: [ConvnetJS]

Visualization

- Learned weights
- **Activations from data**
- Representation space
- Deconvolution-based
- Optimization-based
- DeepDream
- Neural Style

Visualize Activations

Visualize image patches that maximally activate a neuron

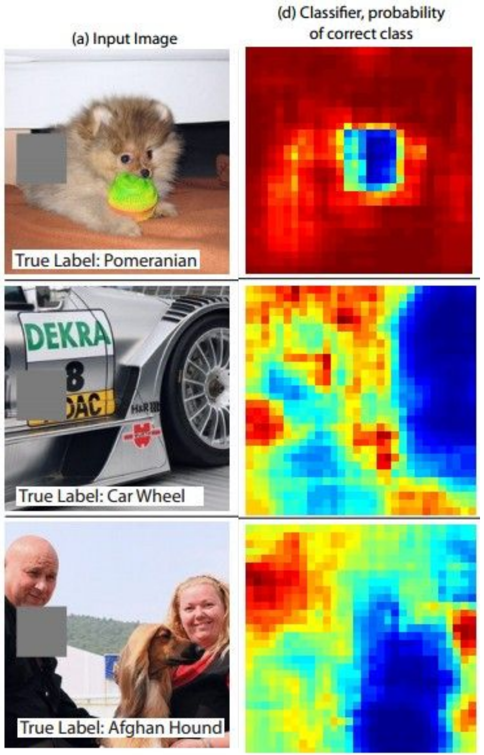
Figure 4: Top regions for six pool$_5$ units. Receptive fields and activation values are drawn in white. Some units are aligned to concepts, such as people (row 1) or text (4). Other units capture texture and material properties, such as dot arrays (2) and specular reflections (6).

Girshick et al. [Rich feature hierarchies for accurate object detection and semantic segmentation.](#) CVPR 2014

Visualize Activations

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

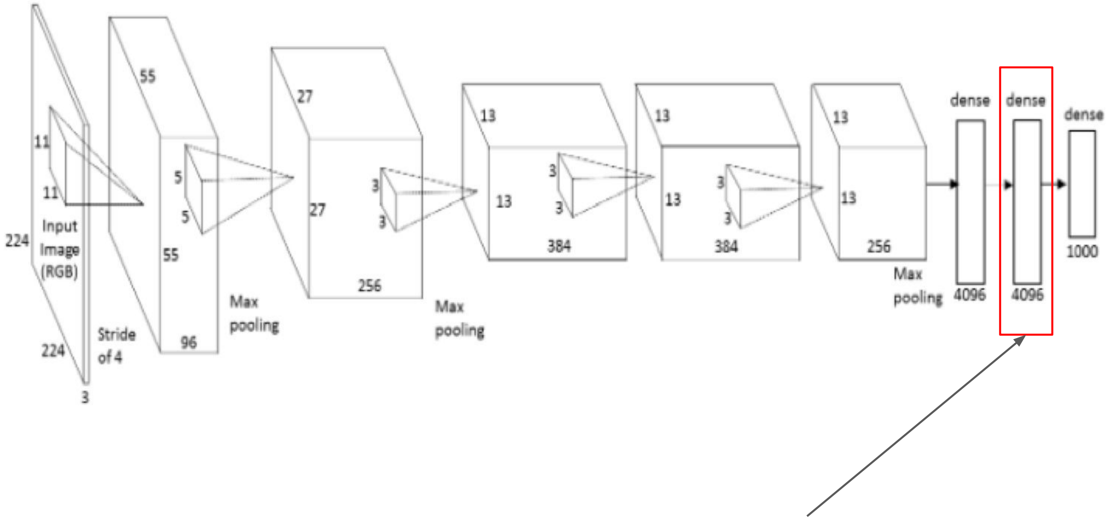


Zeiler and Fergus. Visualizing and Understanding Convolutional Networks. ECCV 2015

Visualization

- Learned weights
- Activations from data
- **Representation space**
- Deconvolution-based
- Optimization-based
- DeepDream
- Neural Style

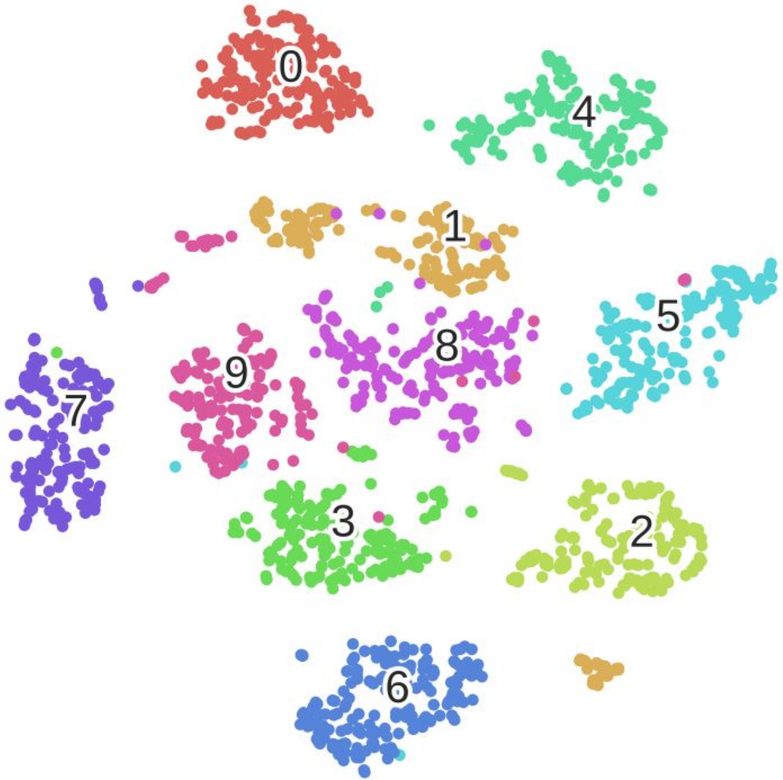
Visualize Representation Space: t-SNE



Extract fc7 as the 4096-dimensional code for each image

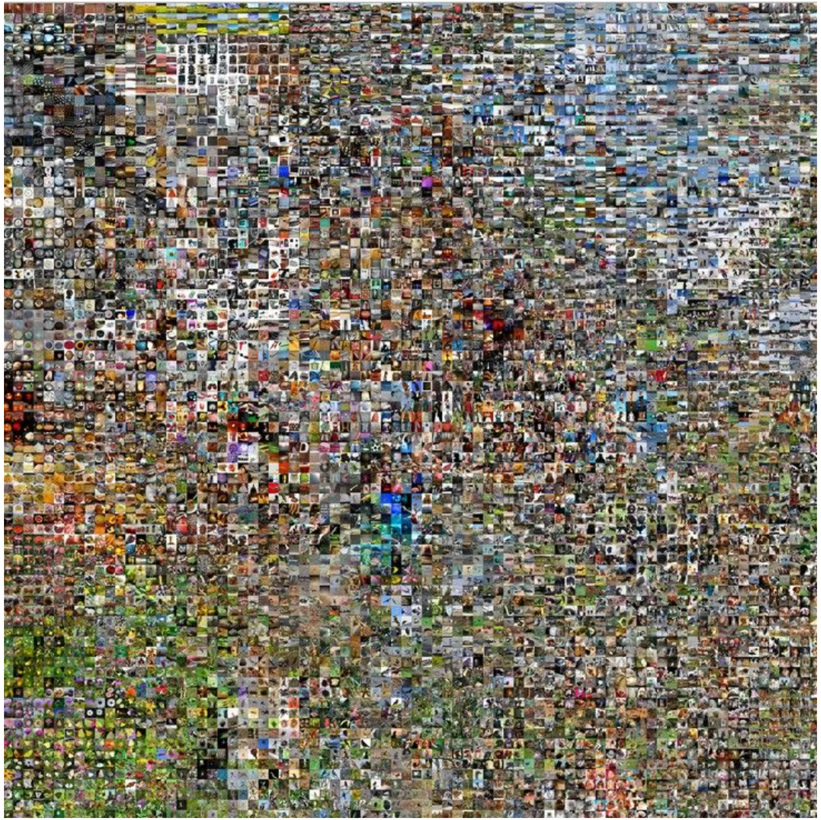
Visualize Representation Space: t-SNE

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



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

Visualize Representation Space: t-SNE



t-SNE on fc7 features from AlexNet.
Source: http://cs.stanford.edu/people/karpathy/cnnembed/

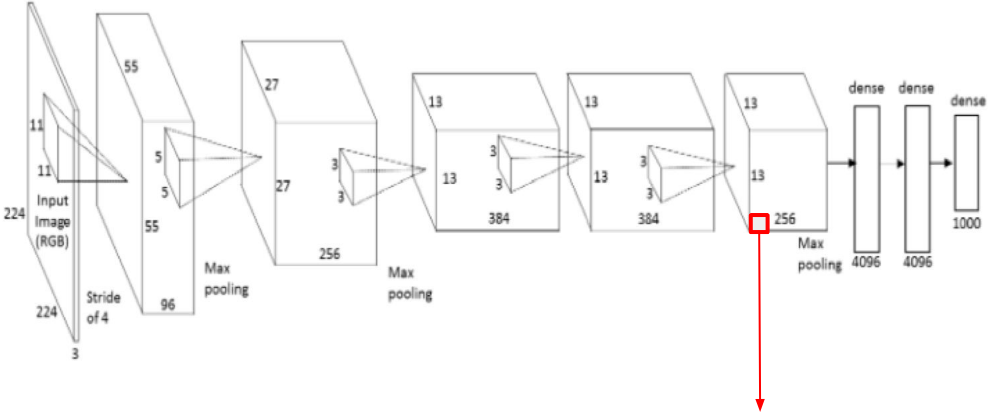
[t-SNE implementation]() on scikit-learn

Visualization

- Learned weights
- Activations from data
- Representation space
- **Deconvolution-based**
- Optimization-based
- DeepDream
- Neural Style

Deconvolution approach

Visualize the part of an image that mostly activates a neuron

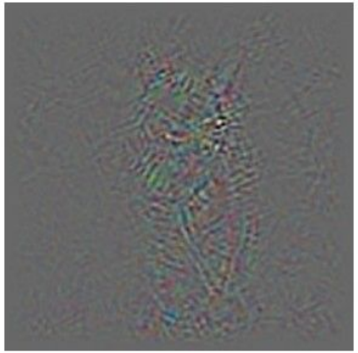


Compute the gradient of any neuron w.r.t. the image

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)

Deconvolution 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)



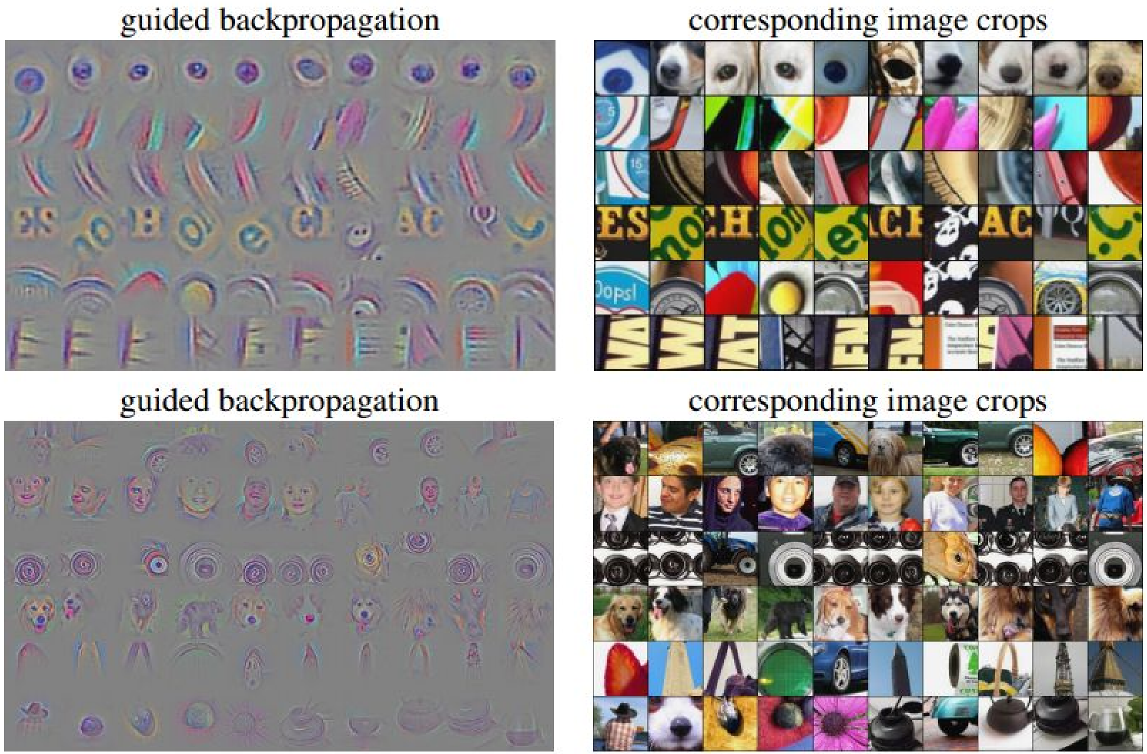
Regular backprop

Guided backprop*

Guided backprop: Only positive gradients are back-propagated. Generates cleaner results.

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

Deconvolution approach



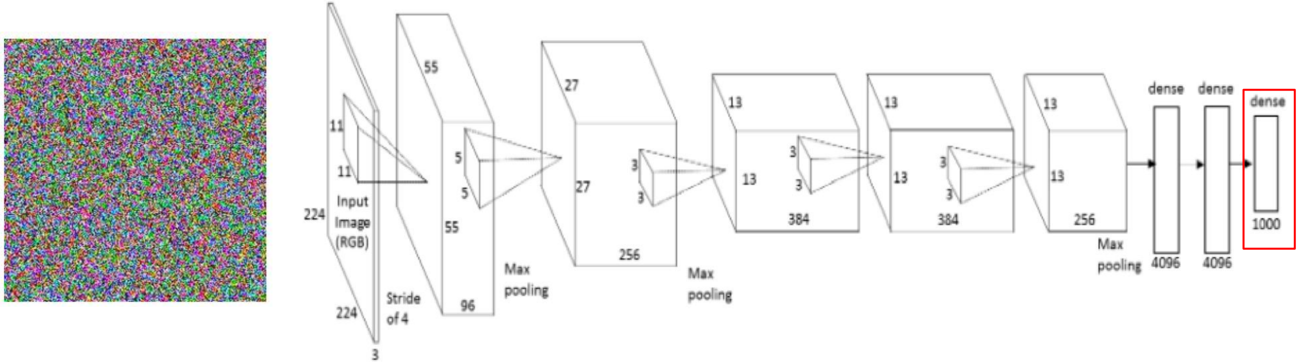
Springenberg, Dosovitskiy, et al. [Striving for Simplicity: The All Convolutional Net](). ICLR 2015

Visualization

- Learned weights
- Activations from data
- Representation space
- Deconvolution-based
- **Optimization-based**
- DeepDream
- Neural Style

Optimization approach

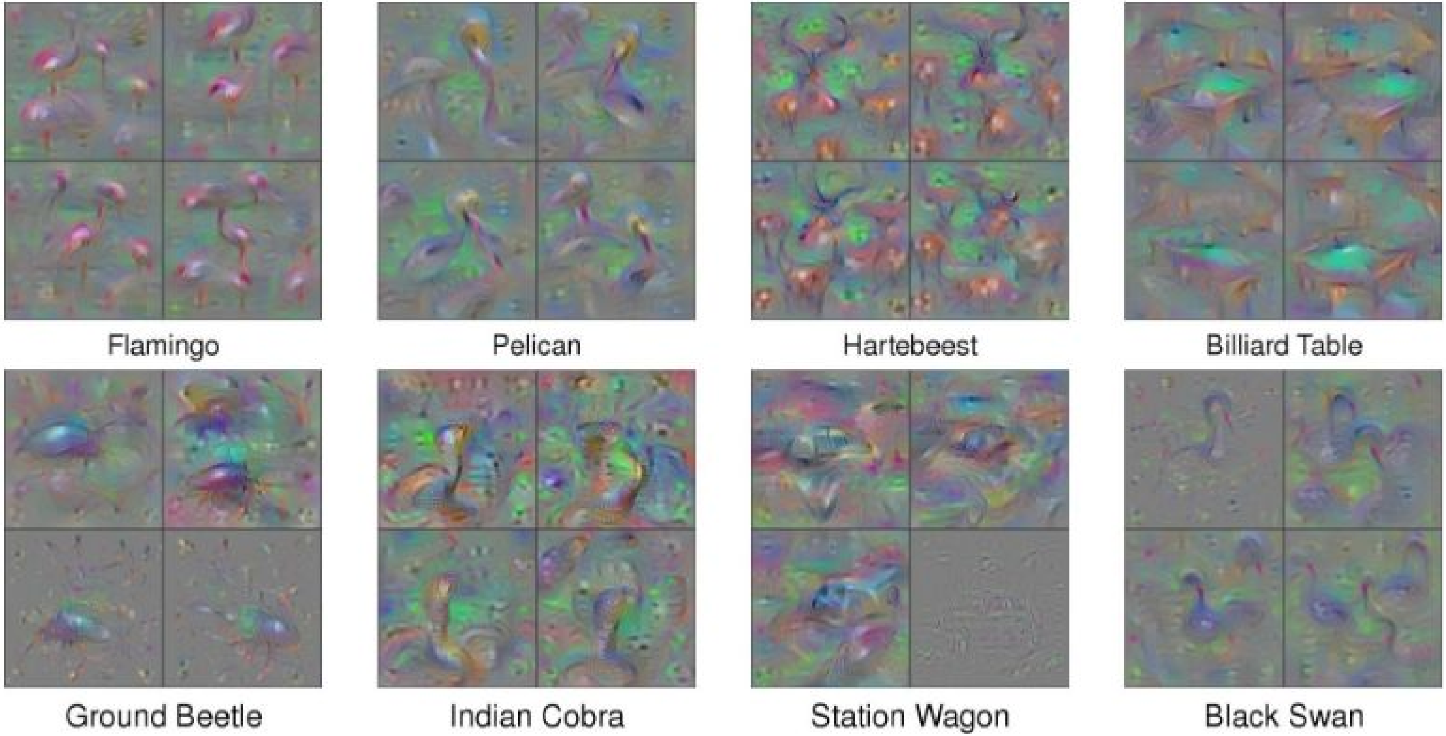
Obtain the image that maximizes a class score



1. Forward random image
2. Set the gradient of the scores vector to be [0,0,0…,1,...,0,0]
3. Backprop (w/ L2 regularization)
4. Update image
5. Repeat

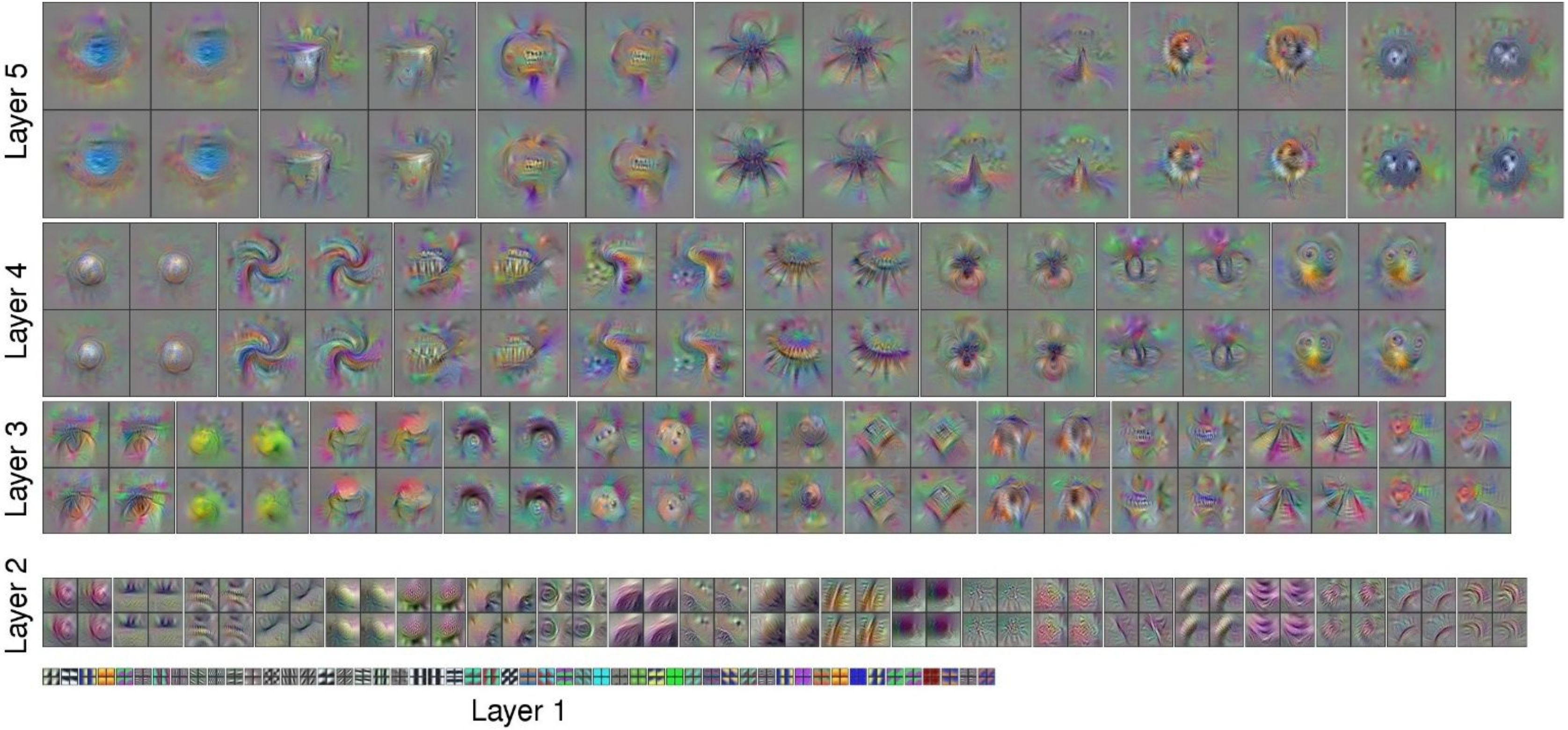
Simonyan et al. [Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps](), 2014

Optimization approach



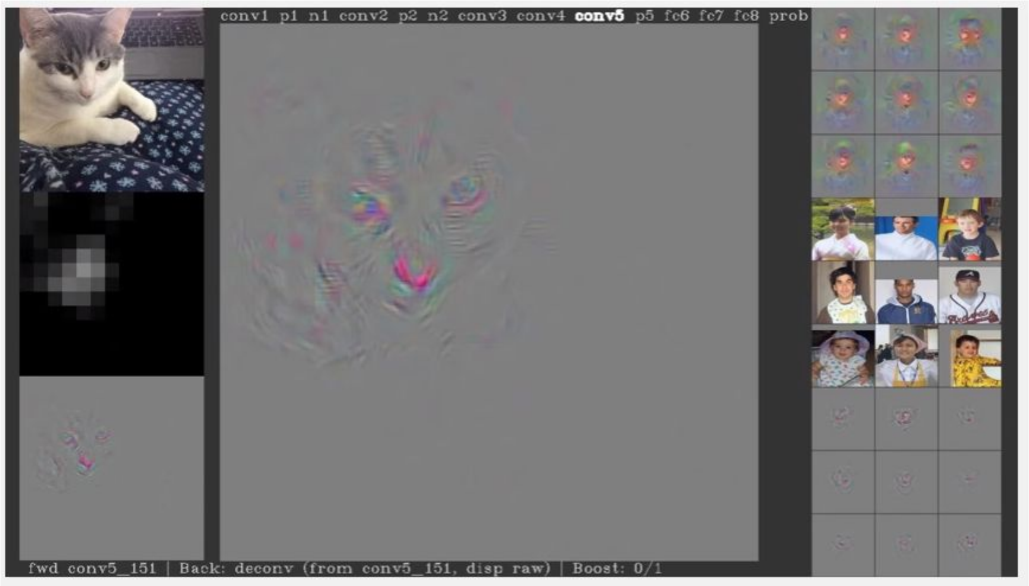
Simonyan et al. [Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps](), 2014

Optimization approach



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

Deep Visualization Toolbox



http://yosinski.com/deepvis

Optimization & Deconv-based visualizations

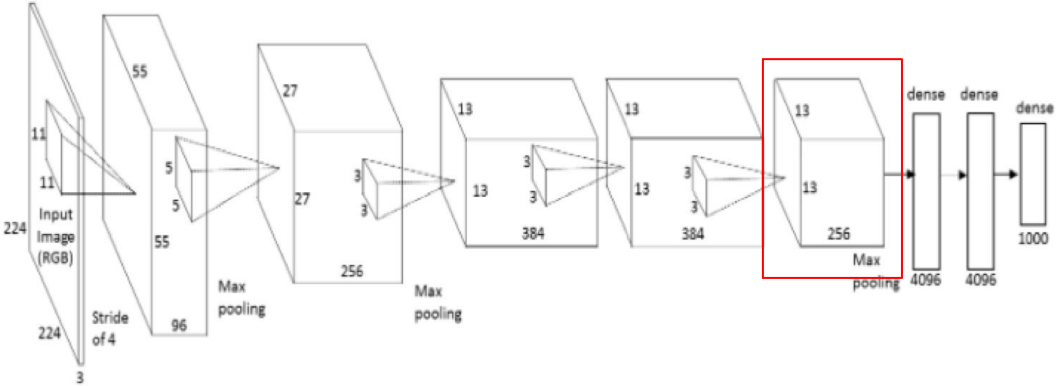
Visualization

- Learned weights
- Activations from data
- Representation space
- Deconvolution-based
- Optimization-based
- **DeepDream**
- Neural Style

DeepDream

https://github.com/google/deepdream

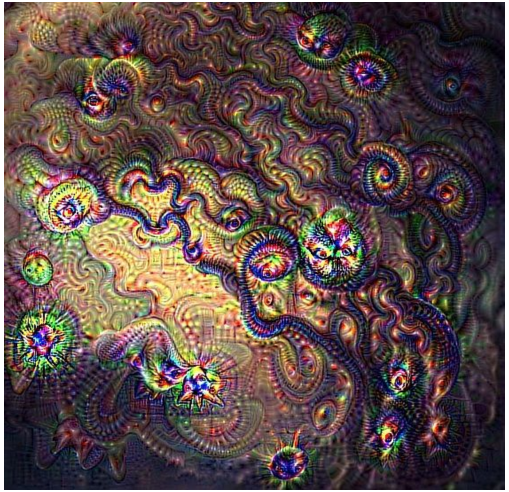
DeepDream



1. Forward image up to some layer (e.g. conv5)
2. Set the gradients to equal the activations on that layer
3. Backprop (with regularization)
4. Update the image
5. Repeat

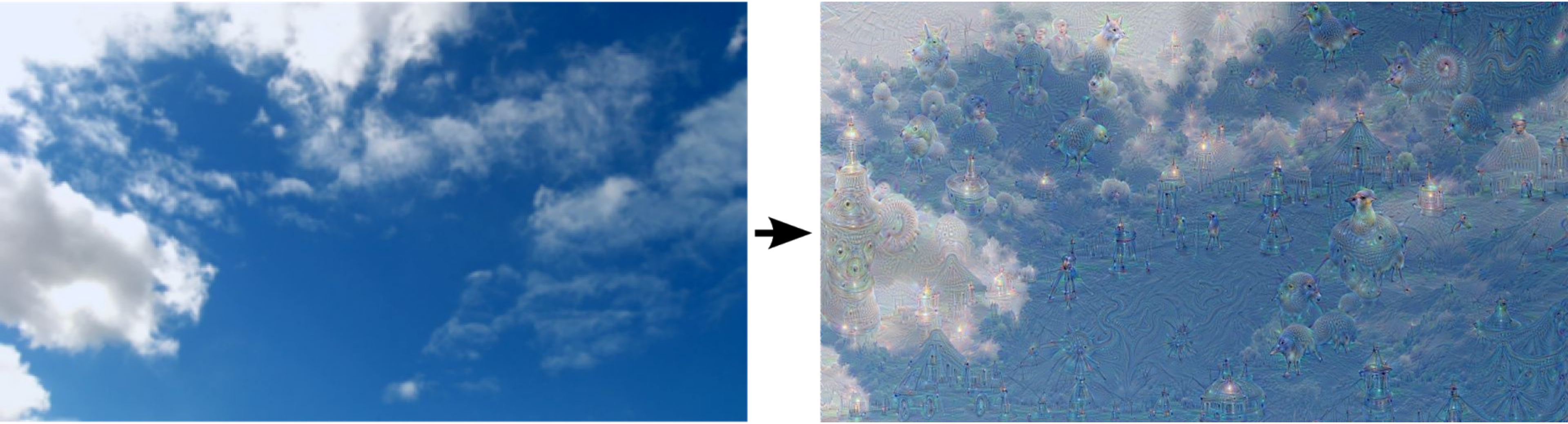
DeepDream

1. Forward image up to some layer (e.g. conv5)
2. **Set the gradients to equal the activations on that layer**
3. Backprop (with regularization)
4. Update the image
5. Repeat



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

DeepDream



More examples [here]

Visualization

- Learned weights
- Activations from data
- Representation space
- Deconvolution-based
- Optimization-based
- DeepDream
- **Neural Style**

Neural Style

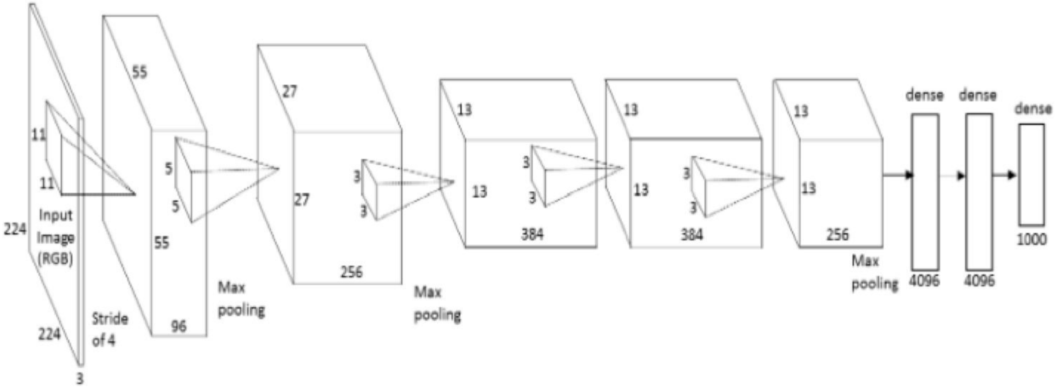
Style image

Content image

Result

Gatys et al. A neural algorithm of artistic style. 2015

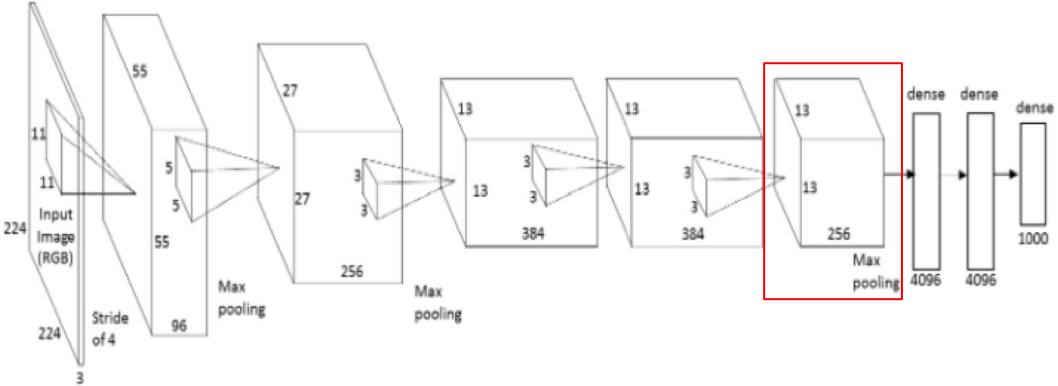
Neural Style



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

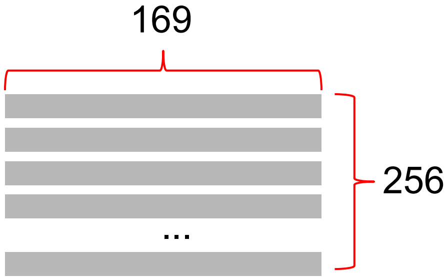
Gatys et al. [A neural algorithm of artistic style](). 2015

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:

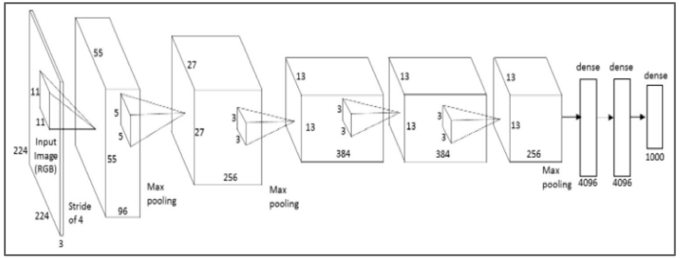
V = 

$G = V^T V$

The Gram matrix G gives the correlations between filter responses.

Neural Style

match content



match style

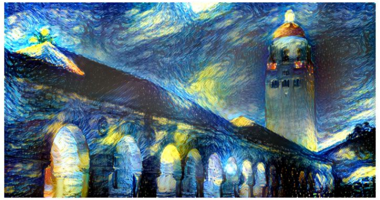
$$\mathcal{L}_{total}(\vec{p},\vec{a},\vec{x}) = \alpha\mathcal{L}_{content}(\vec{p},\vec{x}) + \beta\mathcal{L}_{style}(\vec{a},\vec{x})$$

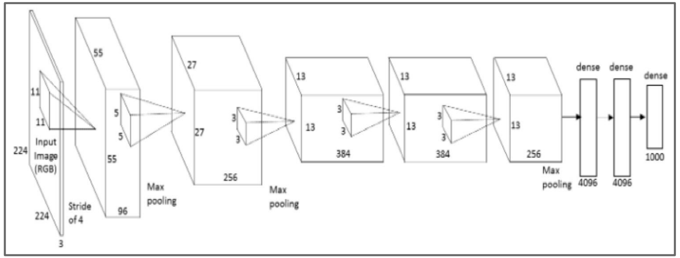
Match activations
from content image

Match gram matrices
from style image

Gatys et al. A neural algorithm of artistic style. 2015

Neural Style





match content

match style

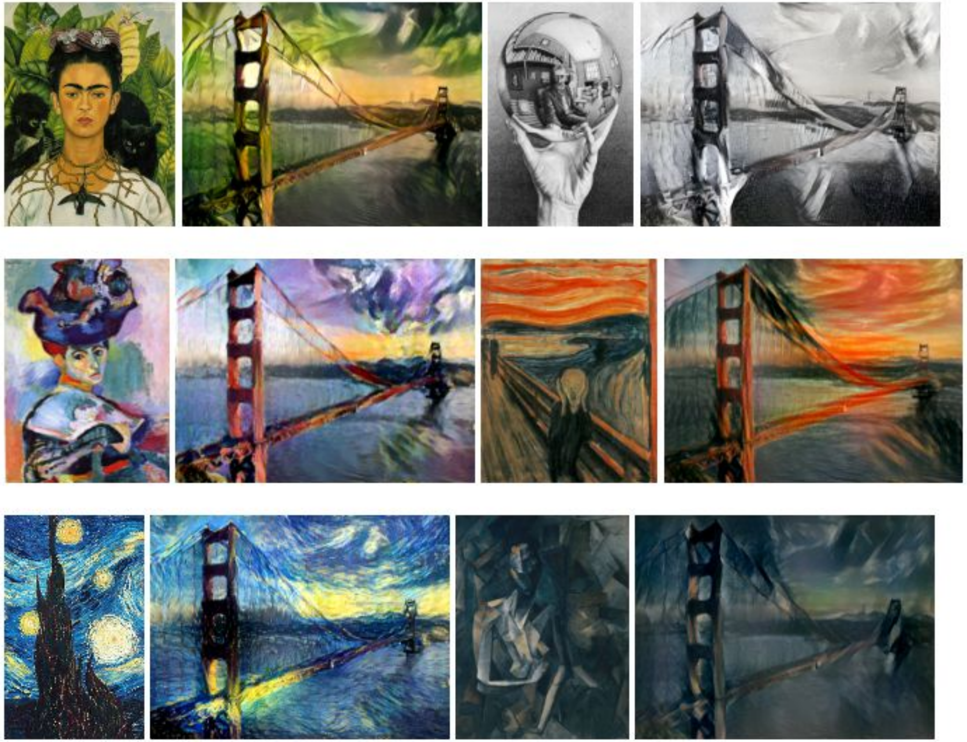
$$\mathcal{L}_{total}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{content}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{style}(\vec{a}, \vec{x})$$

Match activations
from content image

Match gram matrices
from style image

Gatys et al. A neural algorithm of artistic style. 2015

Neural Style



Gatys et al. [A neural algorithm of artistic style](). 2015

Visualization

- Learned weights
- Activations from data
- Representation space
- Deconvolution-based
- Optimization-based
- DeepDream
- Neural Style

Resources

- Related Lecture from CS231n @ Stanford [slides][video]
- ConvnetJS
- t-SNE visualization of CNN codes
- t-SNE implementation on scikit-learn
- Deepvis toolbox
- DrawNet from MIT: Visualize strong activations & connections between units
- 3D Visualization of a Convolutional Neural Network
- NeuralStyle:
  - Torch implementation
  - Deepart.io: Upload image, choose style, (wait), download new image with style :)
- Keras examples:
  - Optimization-based visualization Example in Keras
  - DeepDream in Keras
  - NeuralStyle in Keras