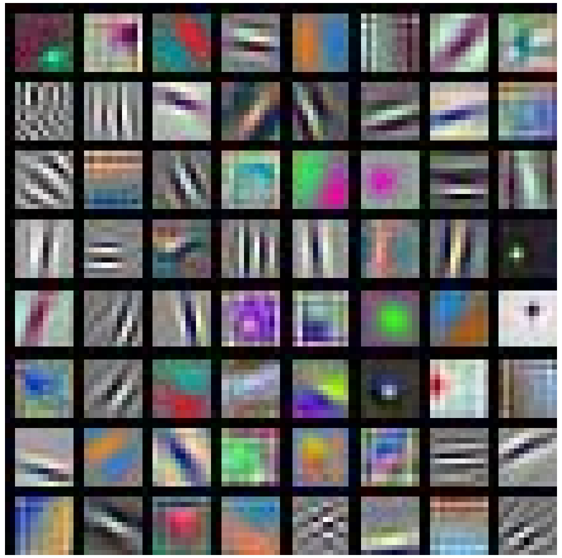
CS231n Lecture 12

- BOAZ 10기 박성현
- BOAZ 11기 김태희
- BOAZ 11기 홍지민
- BOAZ 10기 김용규

01

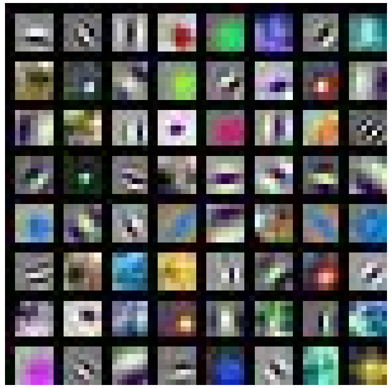
Visualize Layer

First Layer: Visualize Filters



AlexNet:
64 x 3 x 11 x 11

ResNet-18:
64 x 3 x 7 x 7



ResNet-101:
64 x 3 x 7 x 7

DenseNet-121:
64 x 3 x 7 x 7

01 Visualize Layer

Last Layer: Nearest Neighbors

4096-dim vector

Recall: Nearest neighbors in pixel space

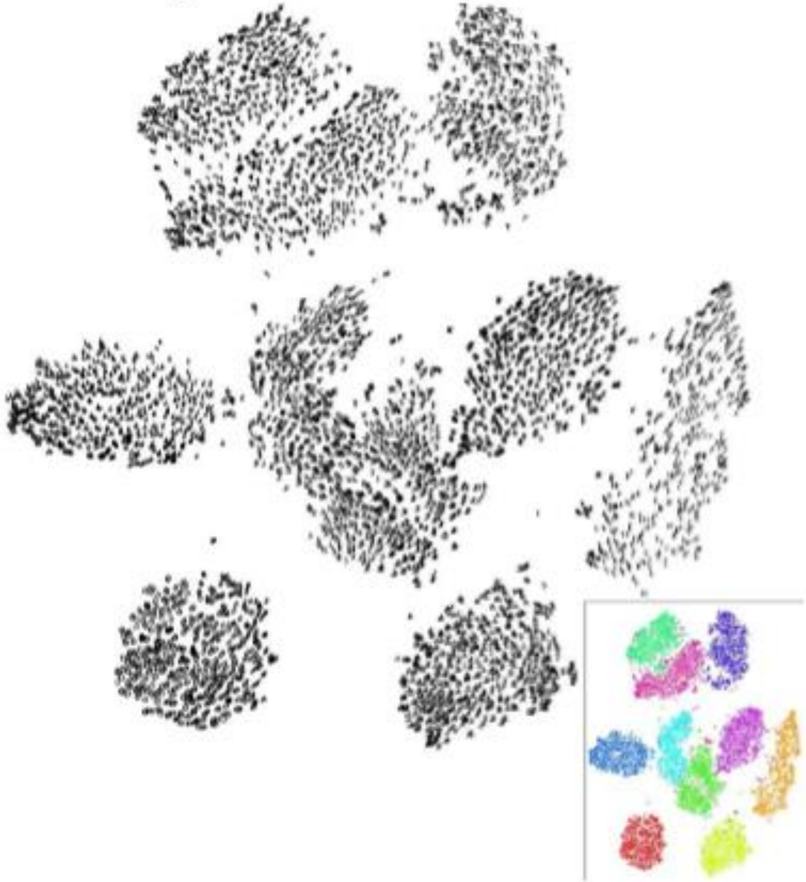
Test image L2 Nearest neighbors in feature space

Last Layer: Dimensionality Reduction

Visualize the “space” of FC7 feature vectors by reducing dimensionality of vectors from 4096 to 2 dimensions

Simple algorithm: Principle Component Analysis (PCA)

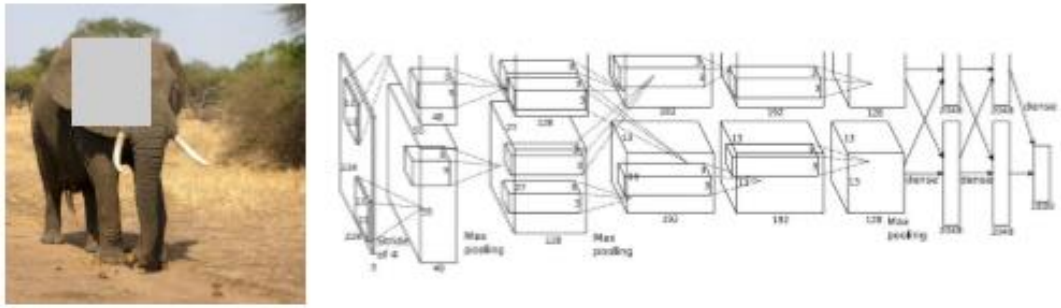
More complex: **t-SNE**



Saliency Maps

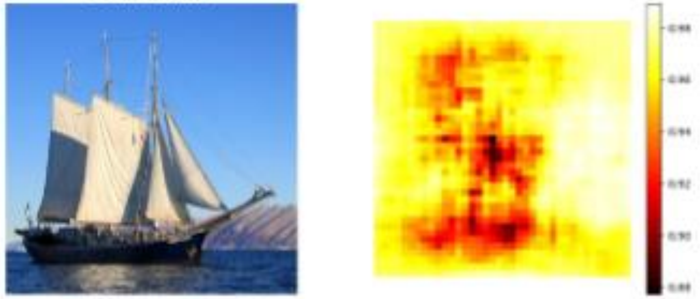
Occlusion Experiments

Mask part of the image before feeding to CNN, draw heatmap of probability at each mask location

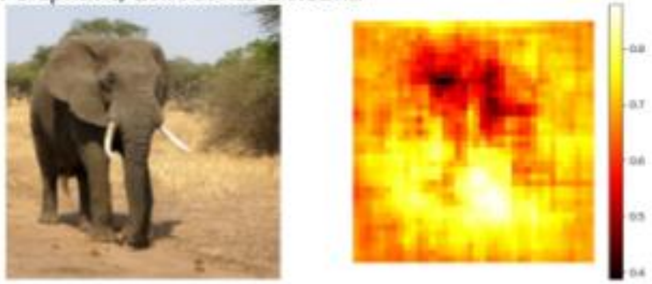


Zeiler and Fergus, "Visualizing and Understanding Convolutional Networks", ECCV 2014

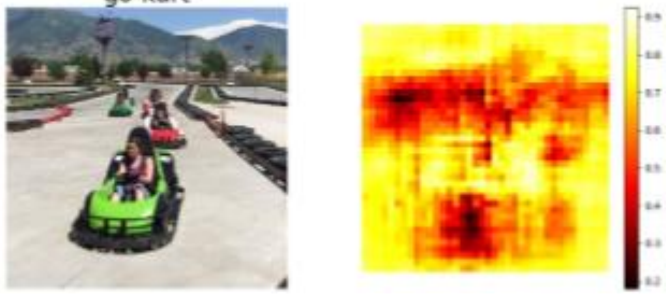
schooner



African elephant, Loxodonta africana



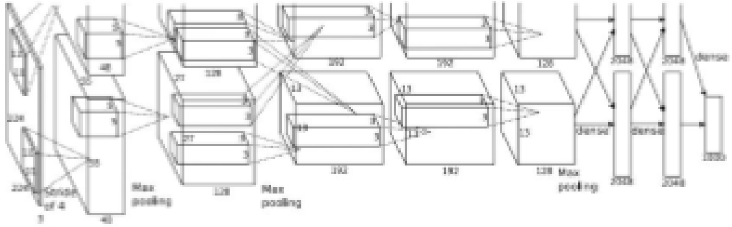
go-kart



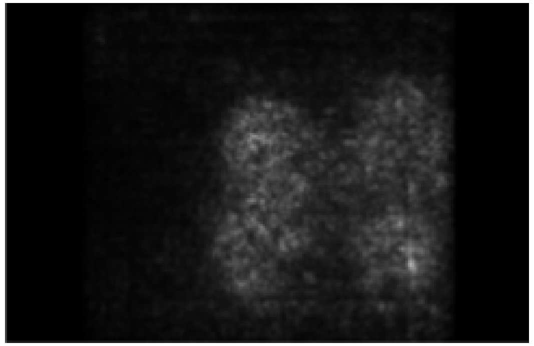
02 Saliency Maps

Saliency Maps

How to tell which pixels matter for classification?



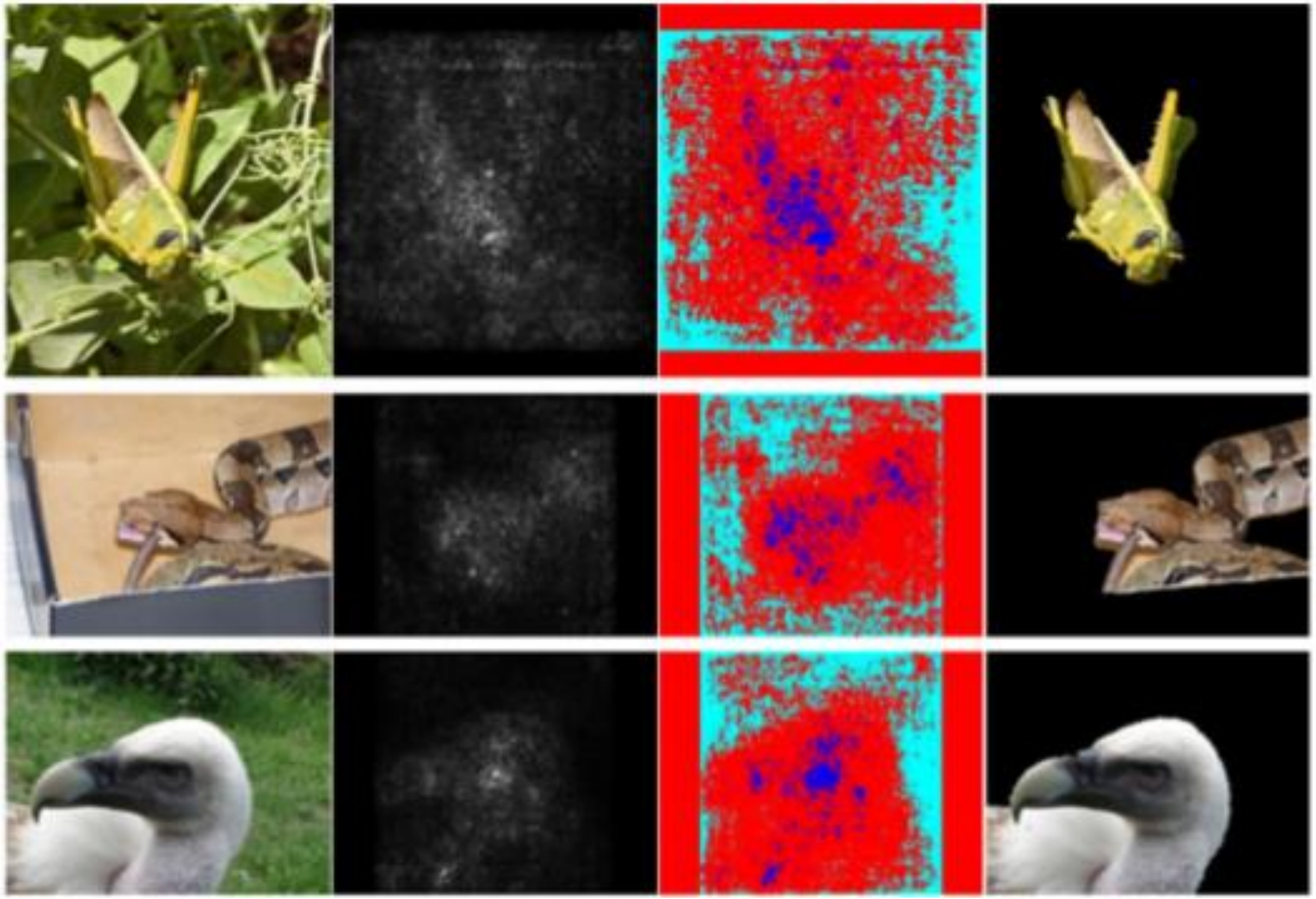
Dog



Compute gradient of (unnormalized) class score with respect to image pixels, take absolute value and max over RGB channels

02

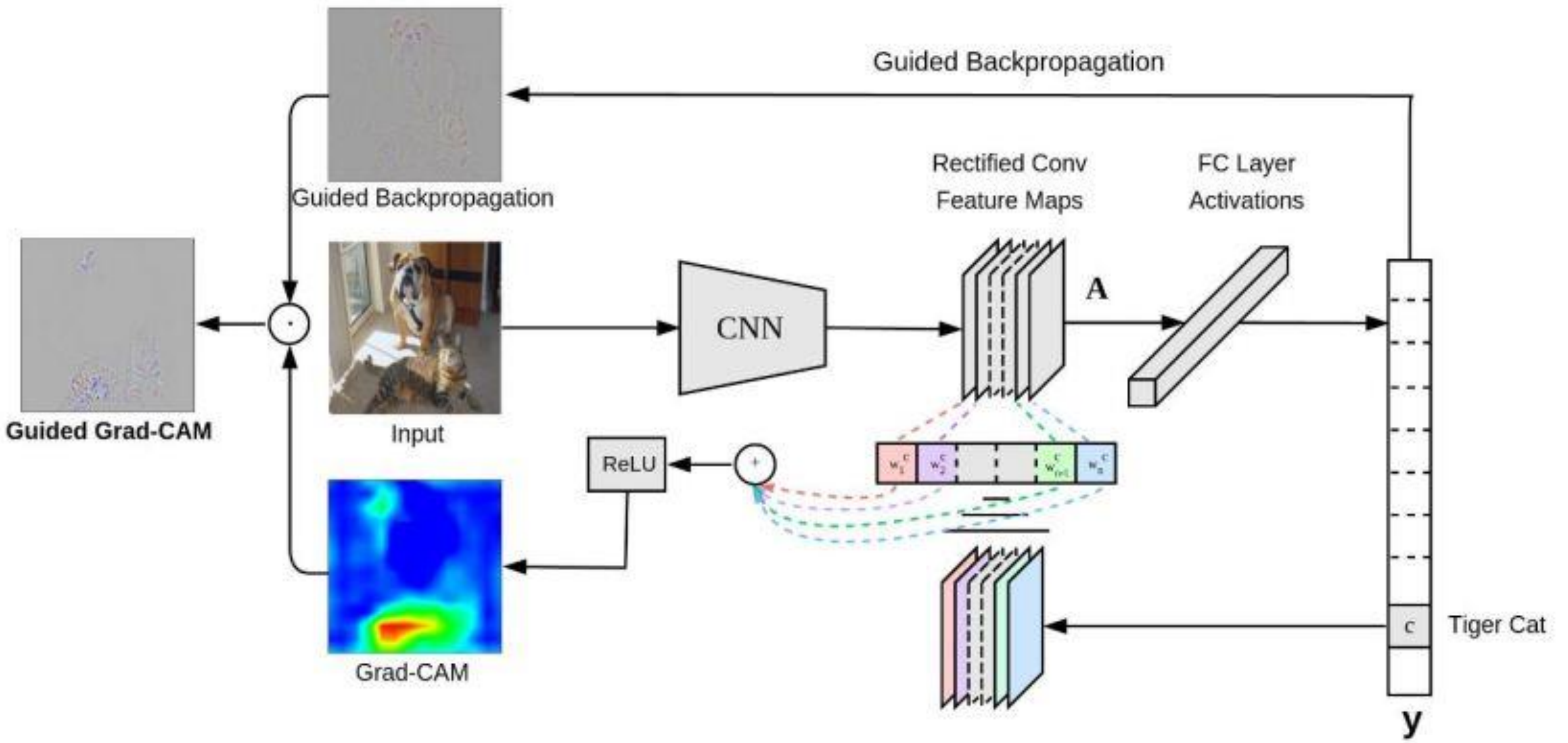
Saliency Maps



03

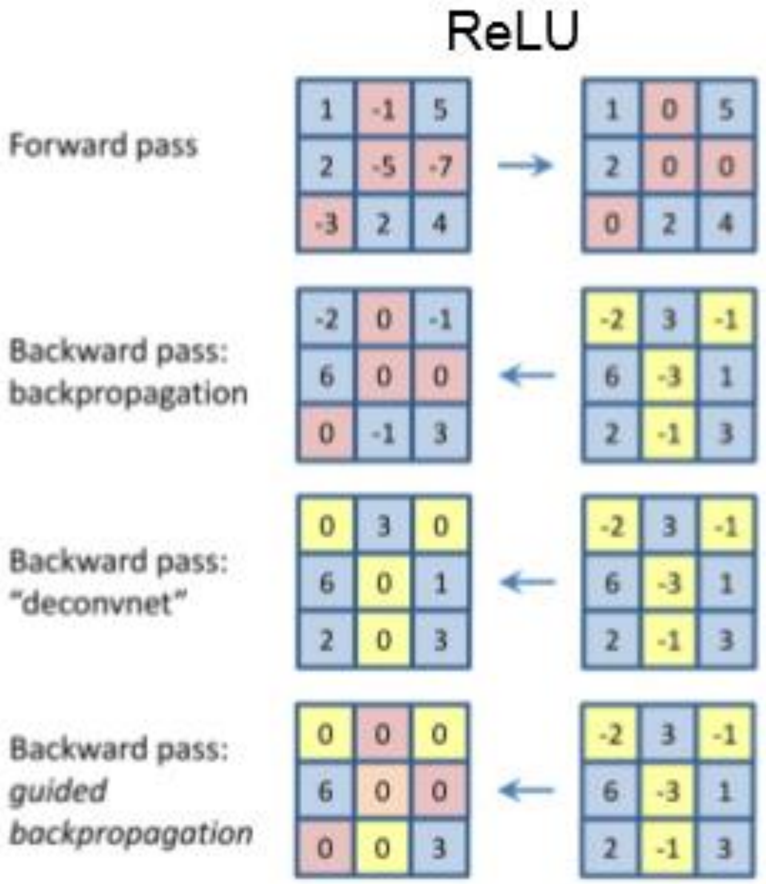
Grad-CAM

03 Grad-CAM



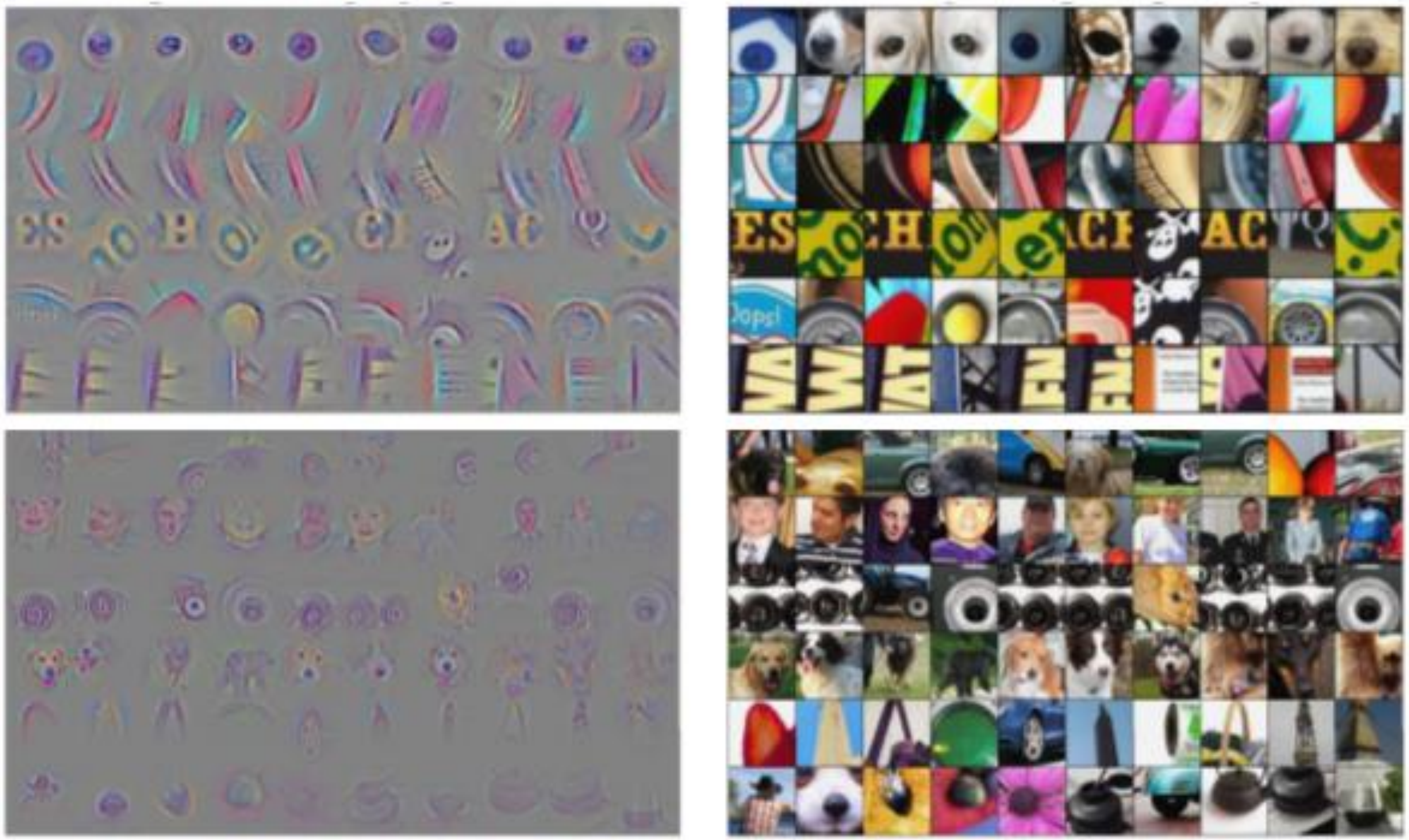
Guided Backpropagation
Guided Backpropagation
Rectified Conv
Feature Maps
FC Layer
Activations
A
CNN
Guided Grad-CAM
Input
ReLU
Tiger Cat
c
Grad-CAM
y

04 Guided Backpropagation



Images come out nicer if you only backprop positive gradients through each ReLU (guided backprop)

04 Guided Backpropagation



Gradient Ascent

(Guided) backprop:
Find the part of an image that a neuron responds to

Gradient ascent:
Generate a synthetic image that maximally activates a neuron

$$I^* = \arg\max_I \boxed{f(I)} + \boxed{R(I)}$$

Neuron value

Natural image regularizer

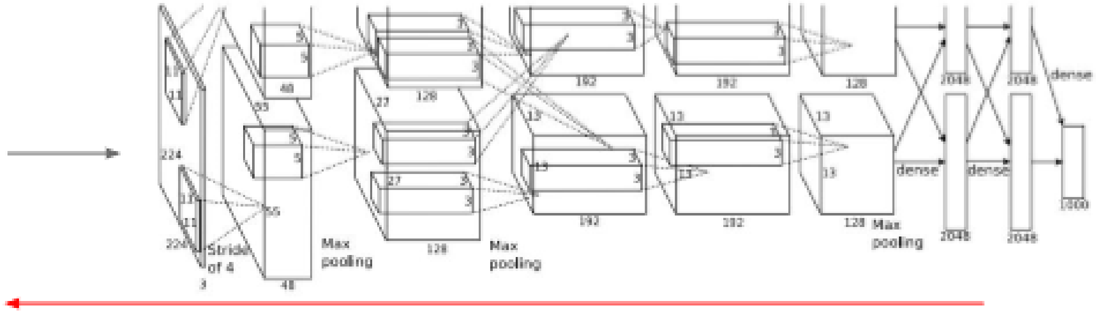
Gradient Ascent

1. Initialize image to zeros

$$\arg\max_{I} \boxed{S_c(I)} - \lambda \|I\|_2^2$$

score for class c (before Softmax)

zero image



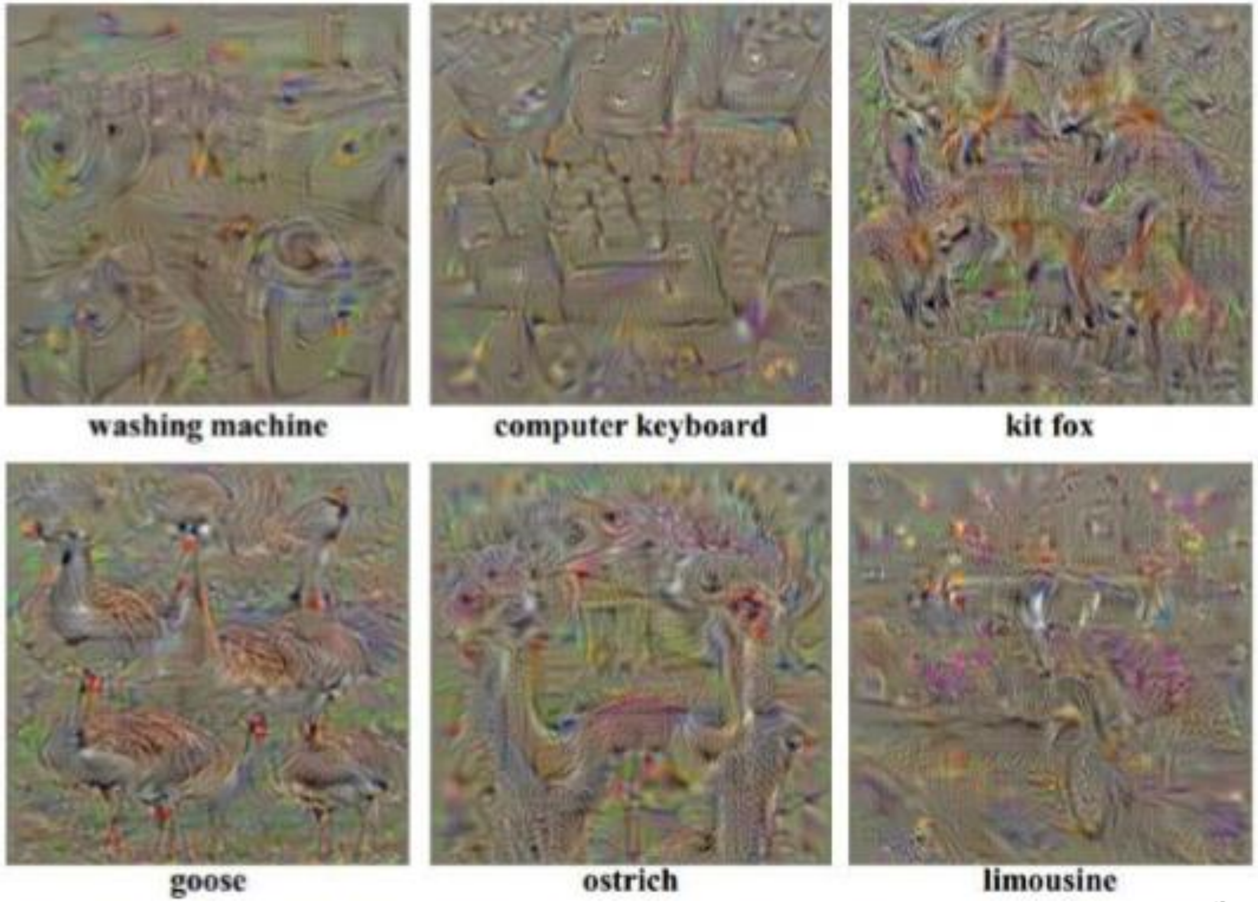
Repeat:

2. Forward image to compute current scores
3. Backprop to get gradient of neuron value with respect to image pixels
4. Make a small update to the image

Gradient Ascent

$$\arg\max_{I} S_c(I) - \boxed{\lambda \|I\|_2^2}$$

Simple regularizer: Penalize L2 norm of generated image

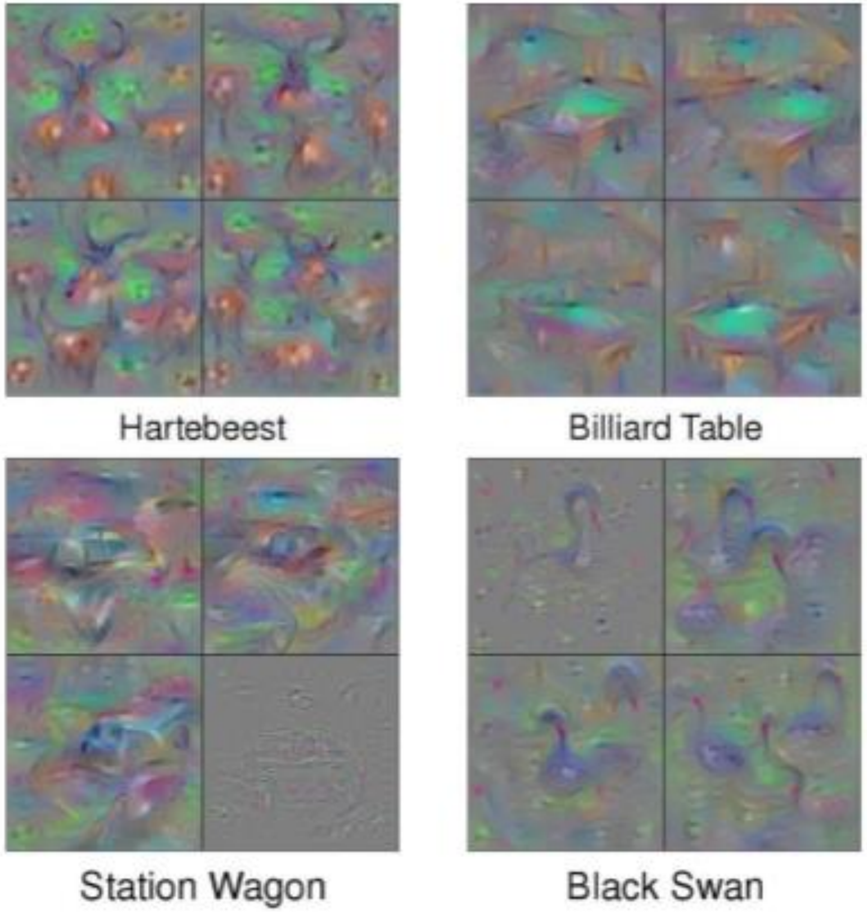


Gradient Ascent

$$\arg\max_I S_c(I) - \lambda \|I\|_2^2$$

Better regularizer: Penalize L2 norm of image; also during optimization periodically

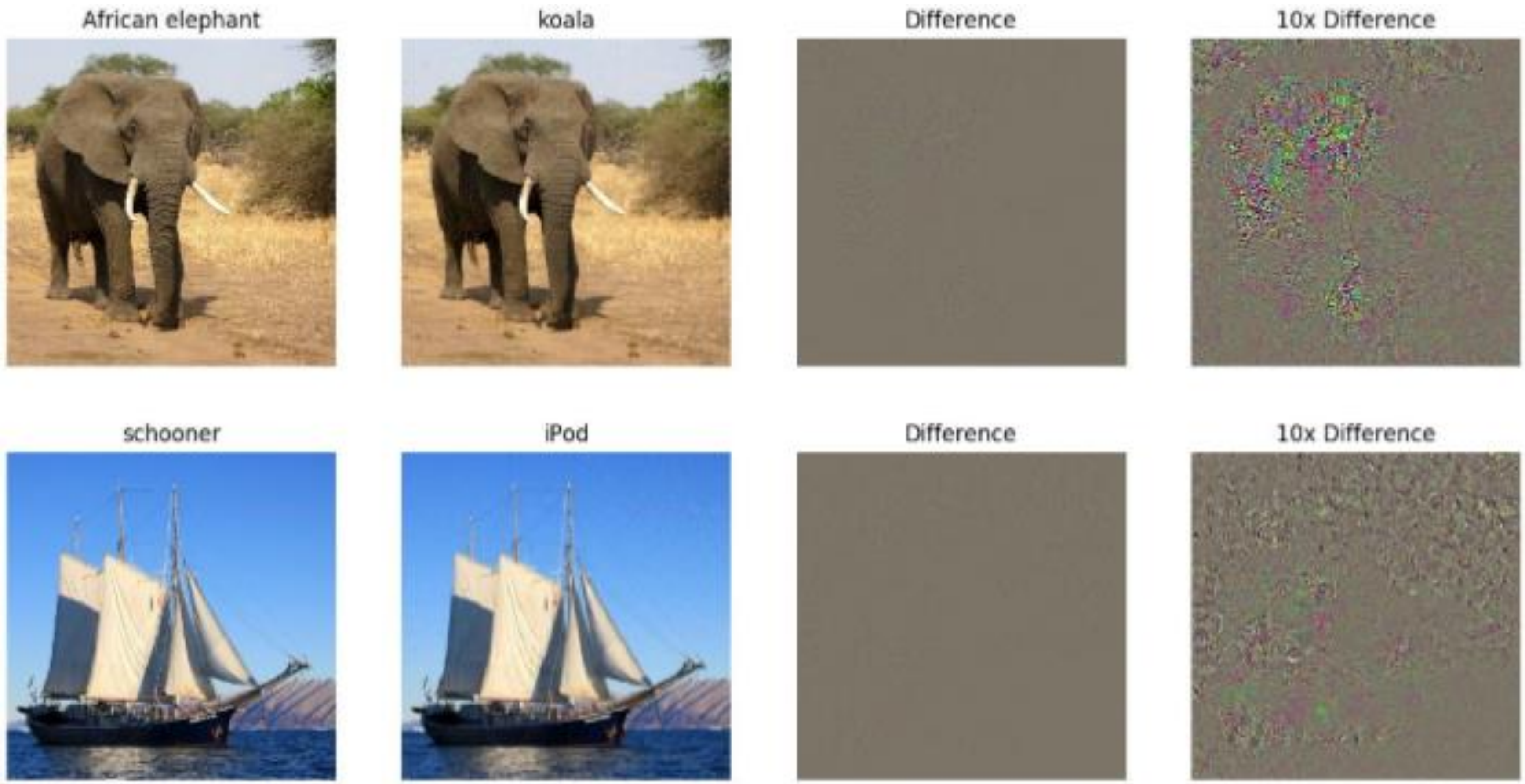
(1) Gaussian blur image
(2) Clip pixels with small values to 0
(3) Clip pixels with small gradients to 0



Fooling Images

(1) Start from an arbitrary image
(2) Pick an arbitrary class
(3) Modify the image to maximize the class
(4) Repeat until network is fooled

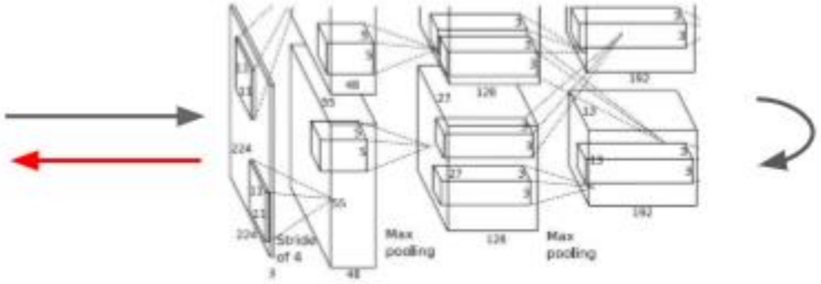
06 Fooling Images



07

DeepDream

Rather than synthesizing an image to maximize a specific neuron, instead try to **amplify** the neuron activations at some layer in the network



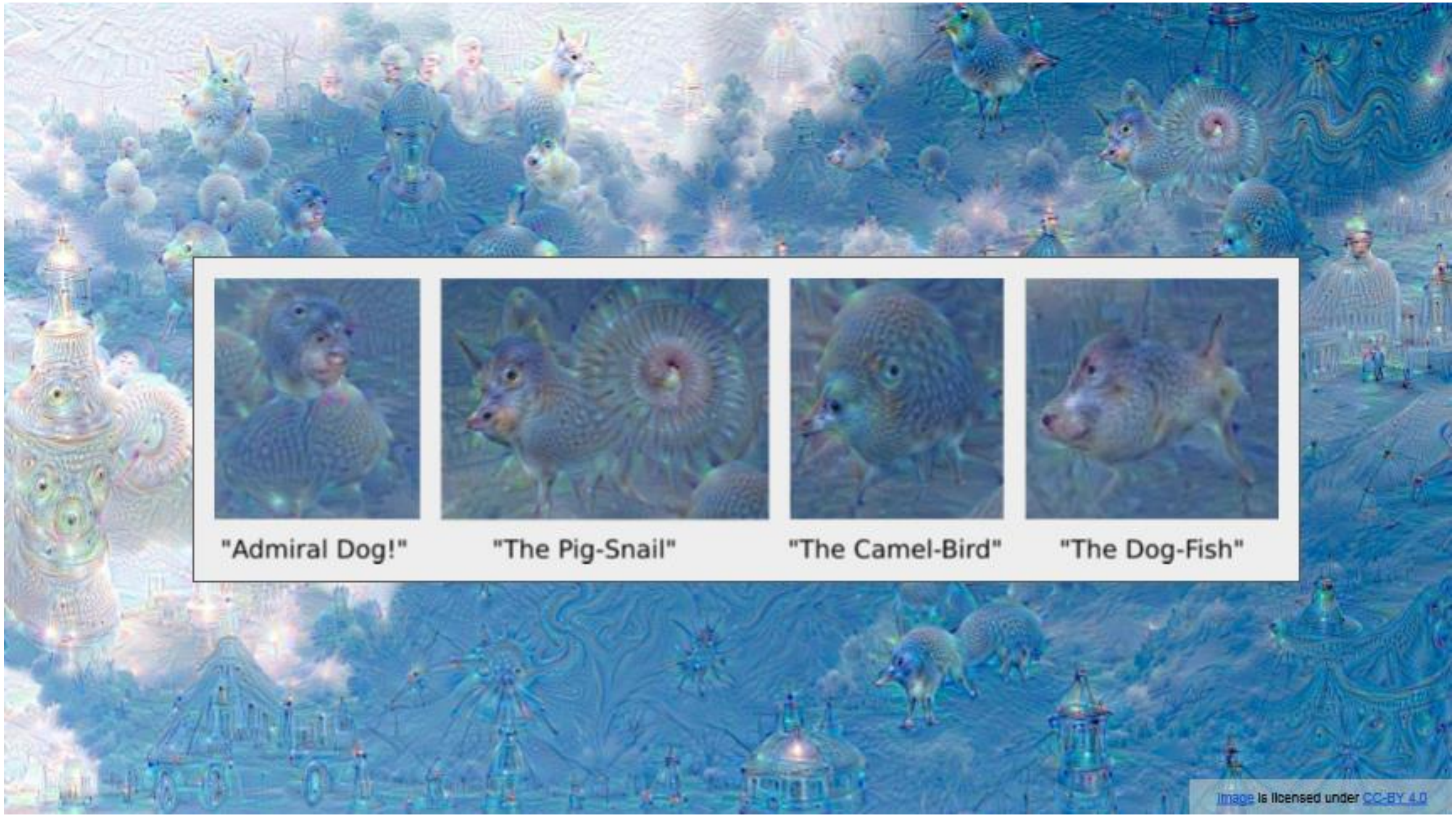
Choose an image and a layer in a CNN; repeat:

1. Forward: compute activations at chosen layer
2. Set gradient of chosen layer *equal to its activation*
3. Backward: Compute gradient on image
4. Update image

Equivalent to:

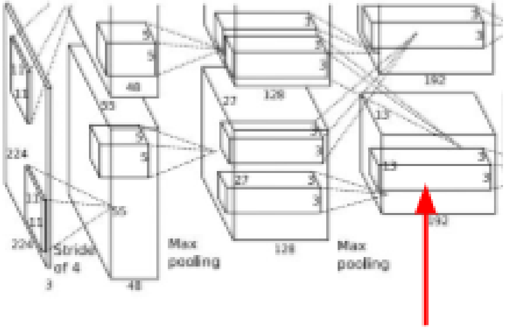
$$I^* = \arg\max_I \sum_i f_i(I)^2$$

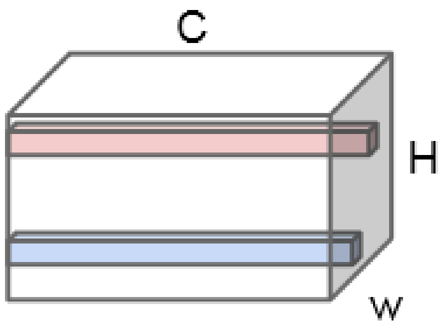
DeepDream

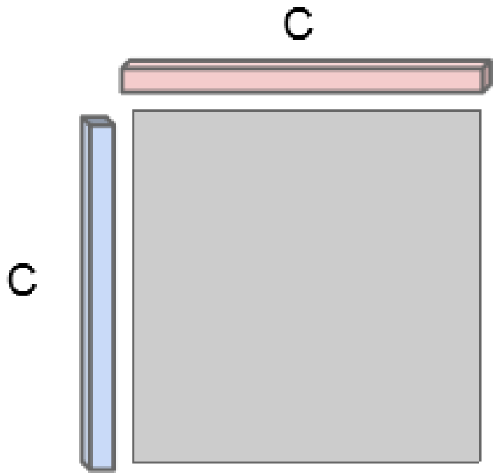

"Admiral Dog!"
"The Pig-Snail"
"The Camel-Bird"
"The Dog-Fish"
Image is licensed under CC-BY 4.0

08 Gram Matrix

This image is in the public domain.





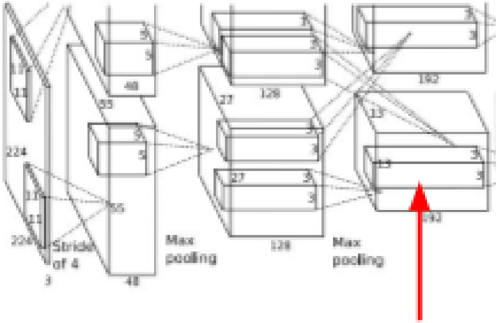


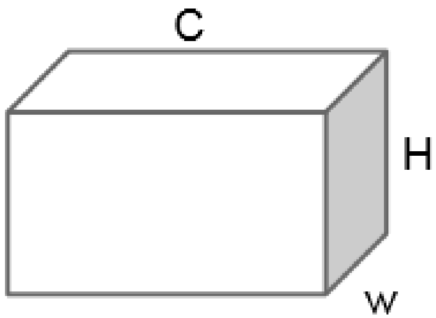
Each layer of CNN gives C x H x W tensor of features; H x W grid of C-dimensional vectors

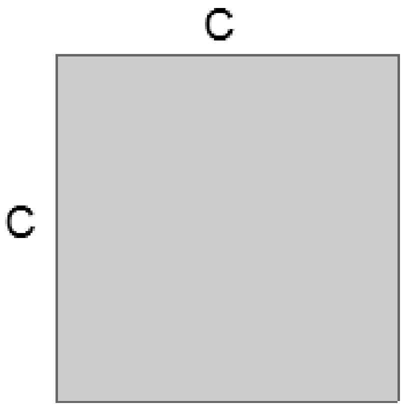
Outer product of two C-dimensional vectors gives C x C matrix measuring co-occurrence

Gram Matrix

This image is in the public domain.







Each layer of CNN gives C x H x W tensor of features; H x W grid of C-dimensional vectors

Outer product of two C-dimensional vectors gives C x C matrix measuring co-occurrence

Average over all HW pairs of vectors, giving **Gram matrix** of shape C x C

Efficient to compute; reshape features from

C x H x W to =C x HW

then compute $G = FF^T$

Style Transfer

Content Image

This image is licensed under CC-BY 3.0

+

Style Image

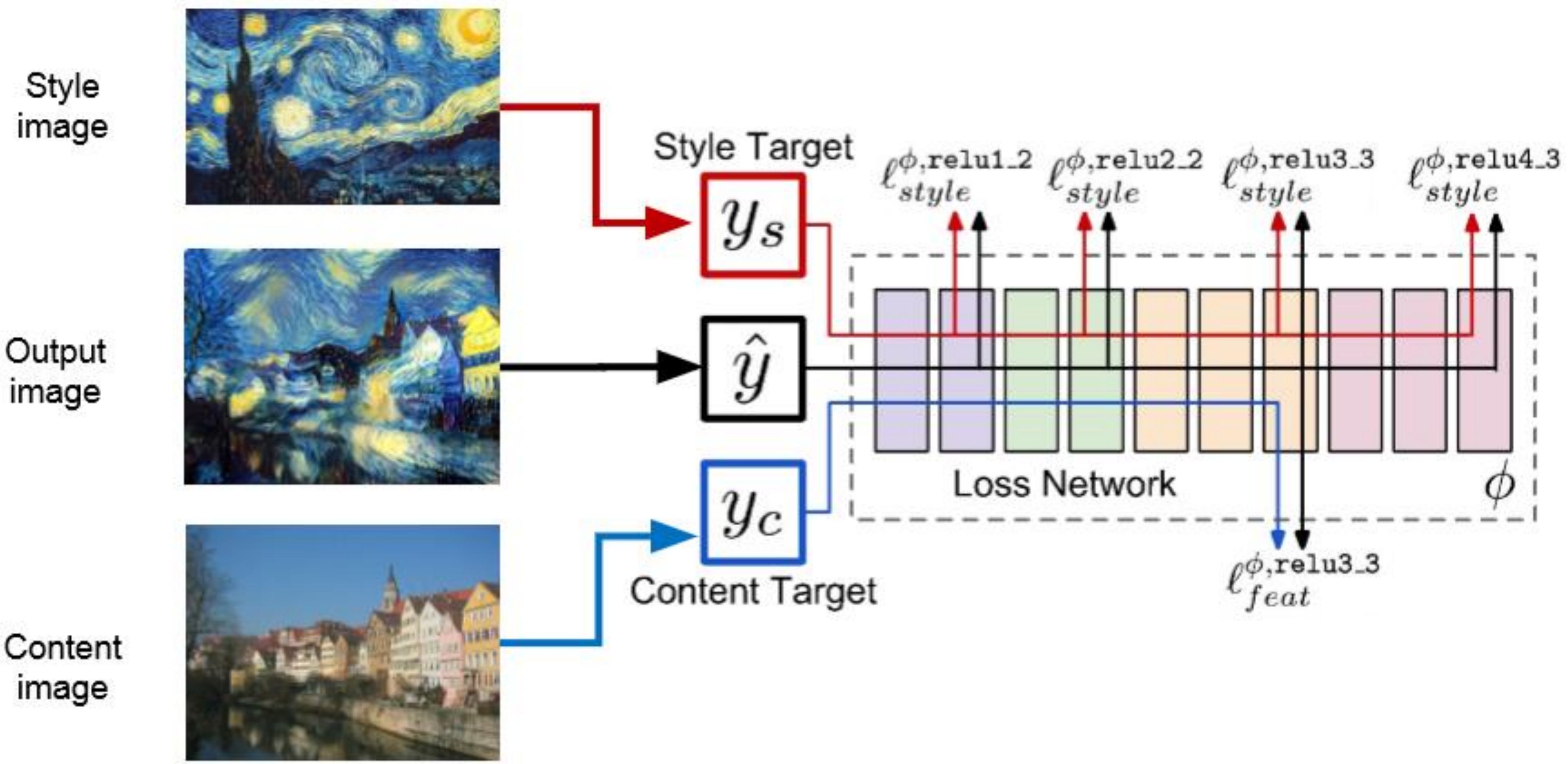
Starry Night by Van Gogh is in the public domain

=

Style Transfer!

This image copyright Justin Johnson, 2015. Reproduced with permission.

09 Style Transfer



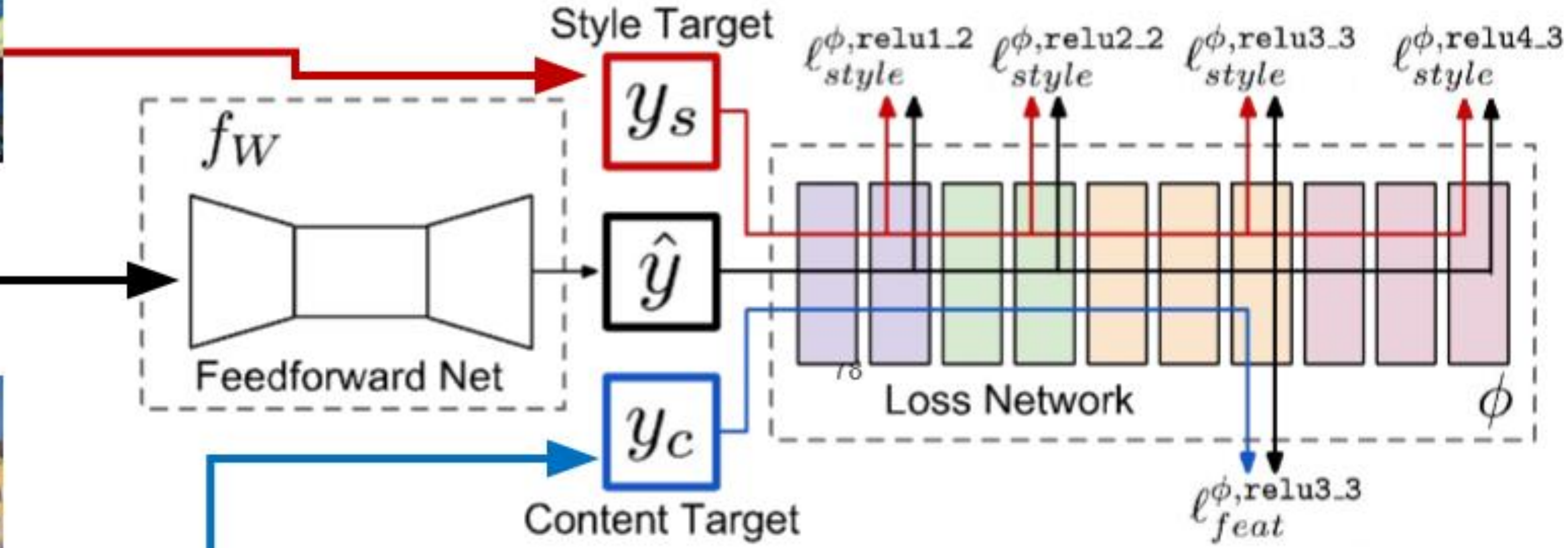
09

Style Transfer

More weight to content loss ⟷ More weight to style loss

Fast Style Transfer

(1) Train a feedforward network for each style
(2) Use pretrained CNN to compute same losses as before
(3) After training, stylize images using a single forward pass



Johnson, Alahi, and Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution", ECCV 2016

End

참고 자료

CS231n : http://cs231n.stanford.edu/syllabus.html

BOAZ