Deep Learning

A course about theory & practice

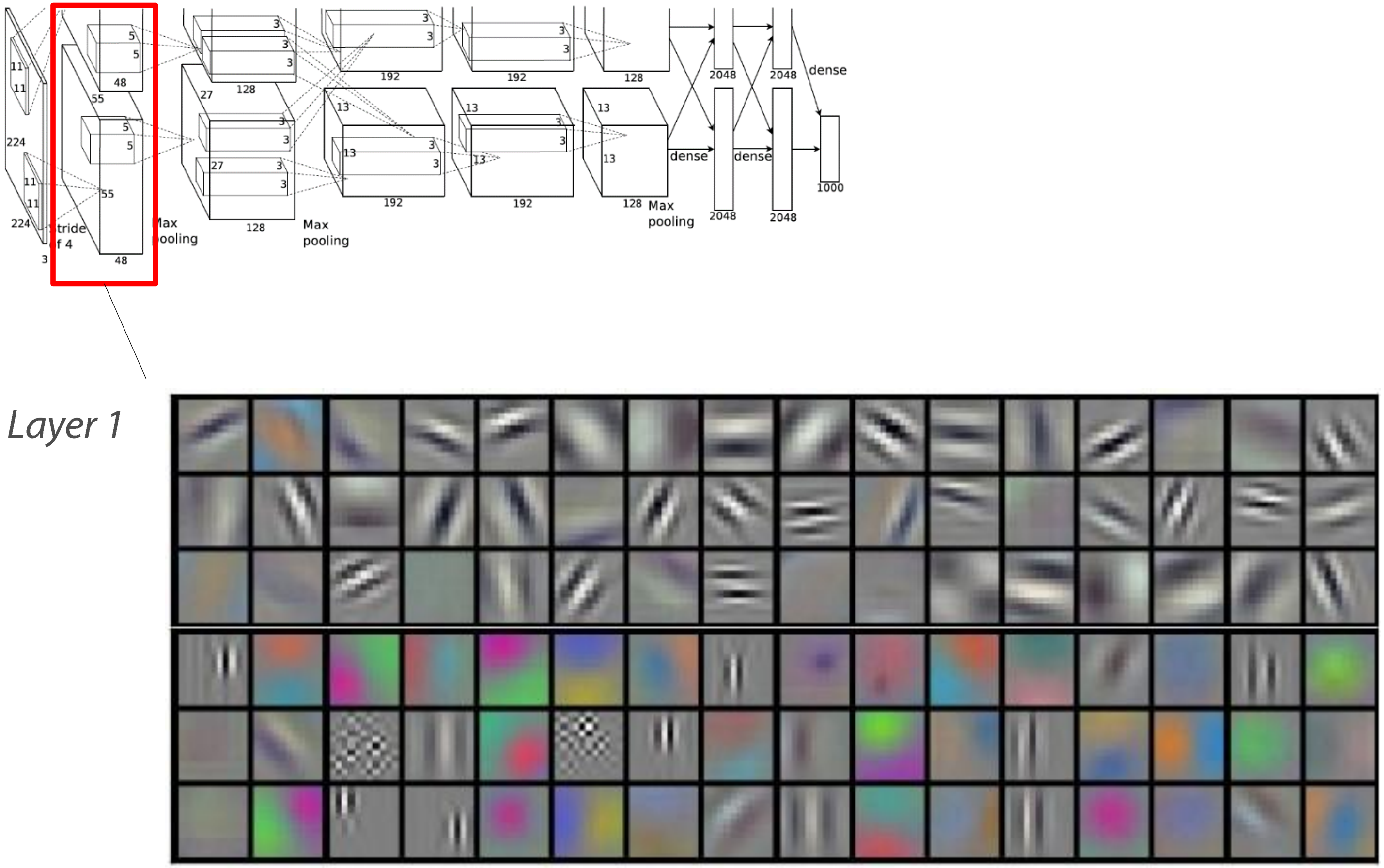
Deep Convolutional Neural Networks *and Beyond*

Marco Piastra

UNIVERSITÀ DI PAVIA

Inside AlexNet (after training)

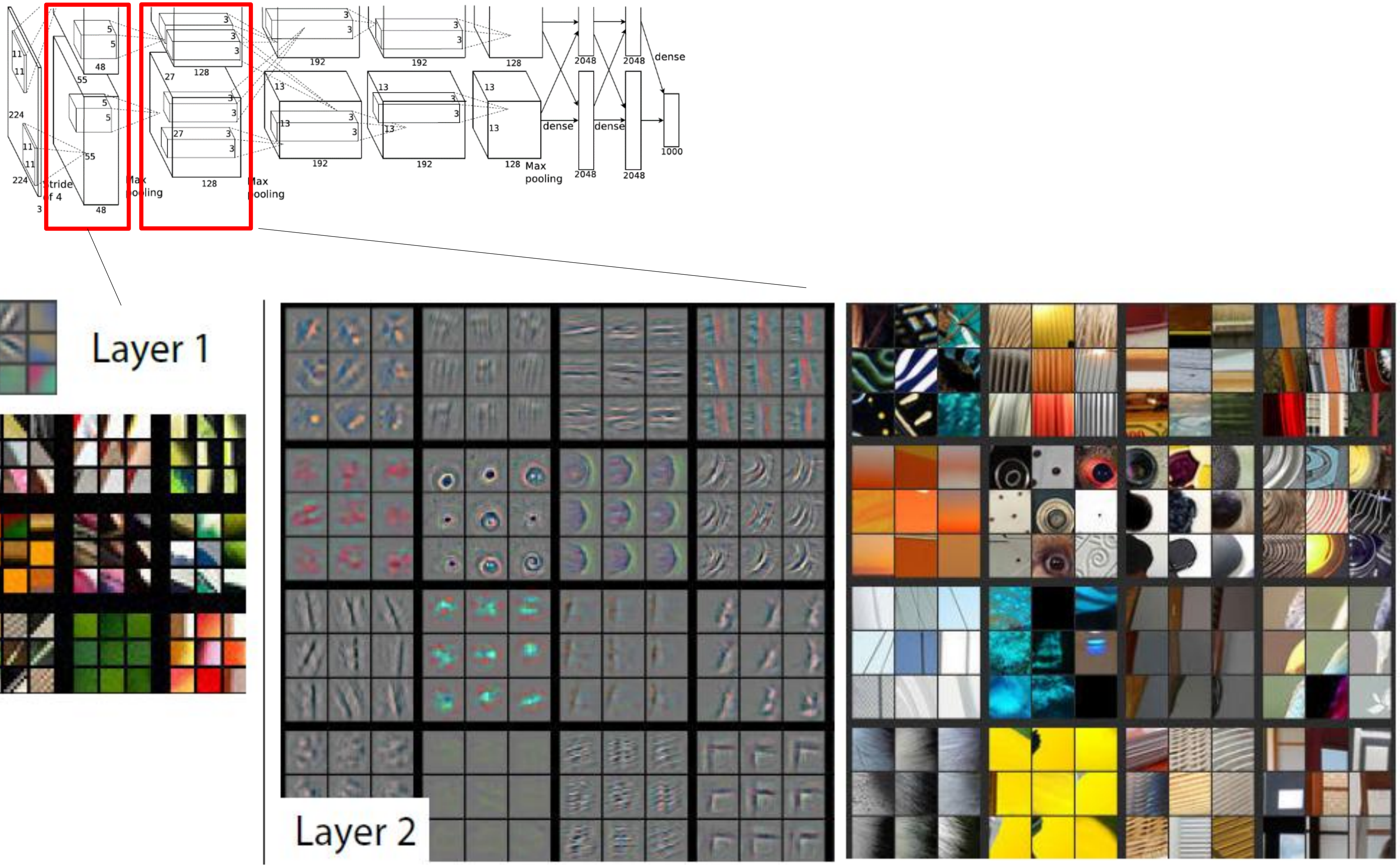
AlexNet Filters (after training)



These are 96 real examples of convolutive filters for RGB images

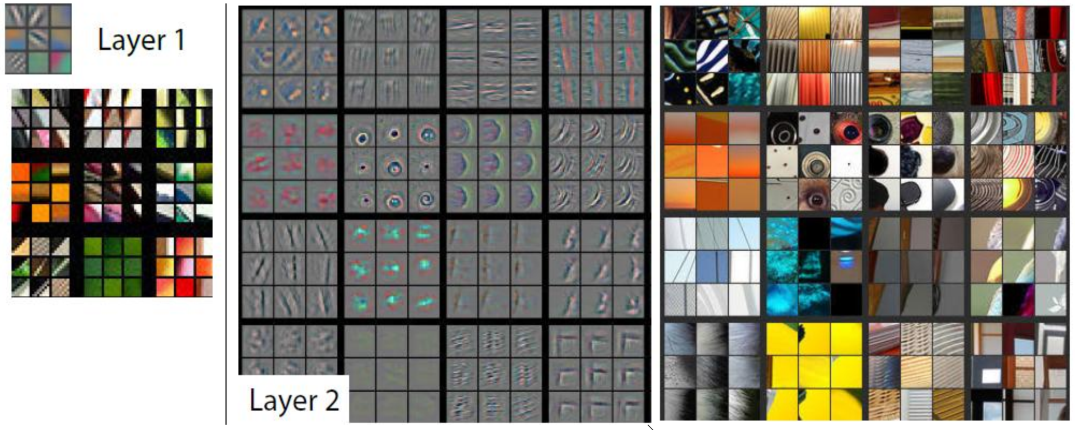
[image from http://cs231n.github.io/convolutional-networks/]

AlexNet Filters- DeconvNet

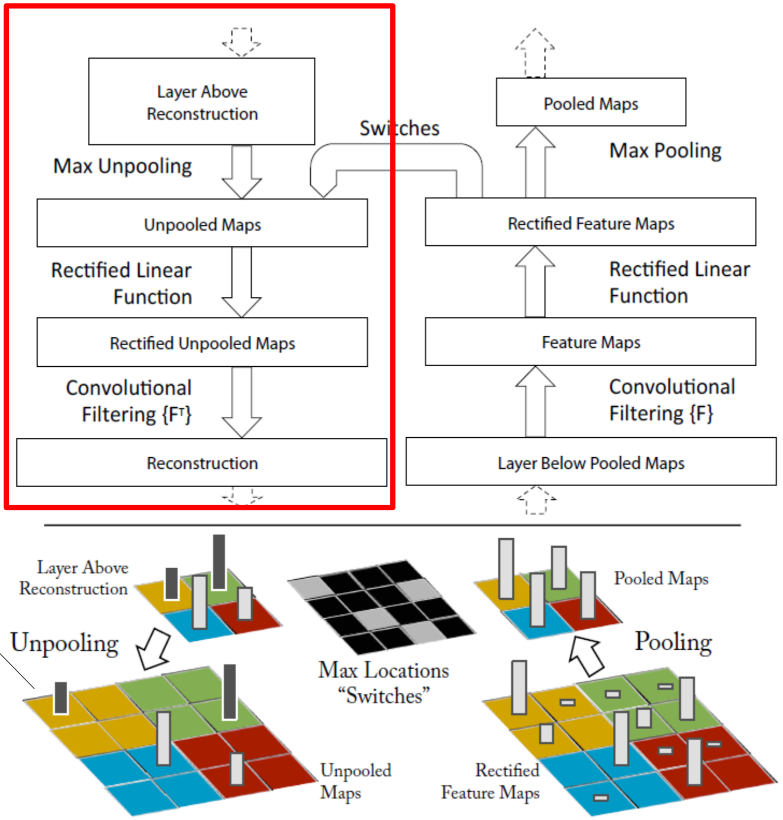


[images from https://arxiv.org/pdf/1311.2901.pdf]

AlexNet Filters- DeconvNet



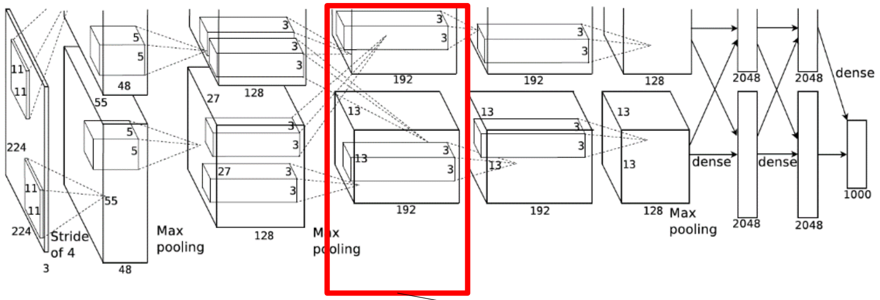
DeconvNet

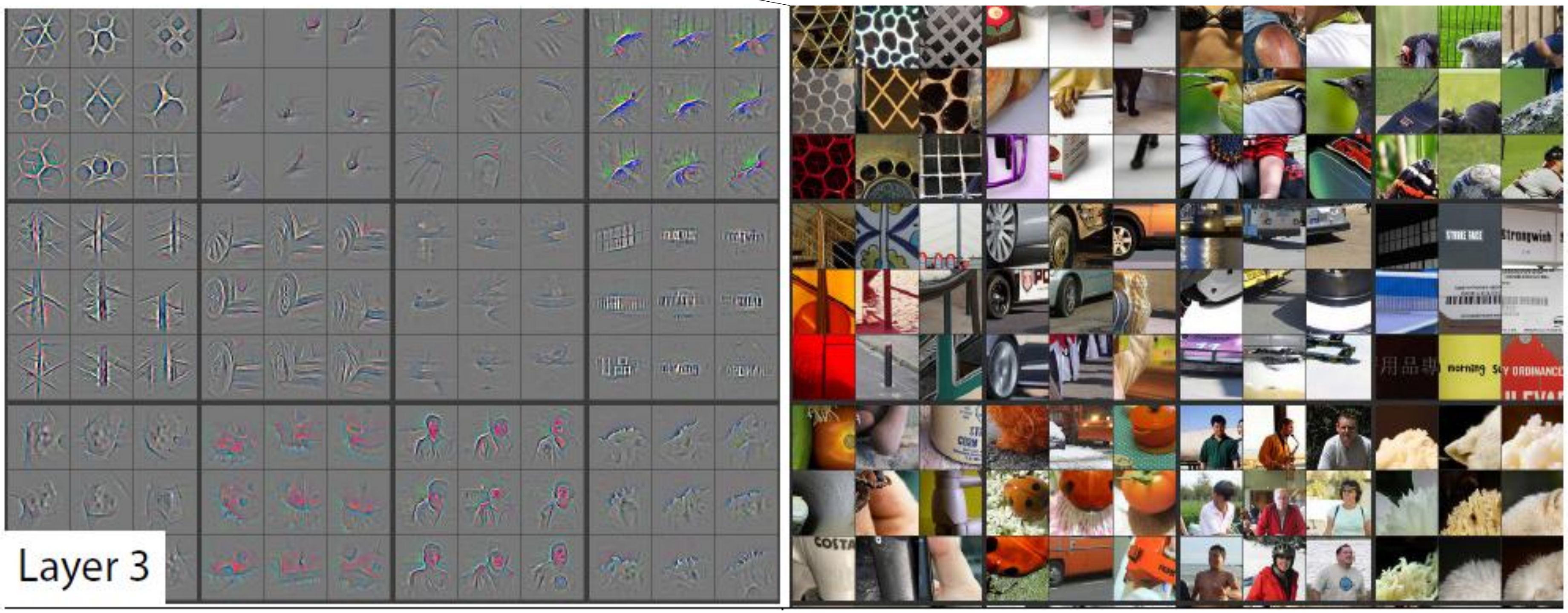


DeconvNet:
using a DCNN in reverse

[images from https://arxiv.org/pdf/1311.2901.pdf]

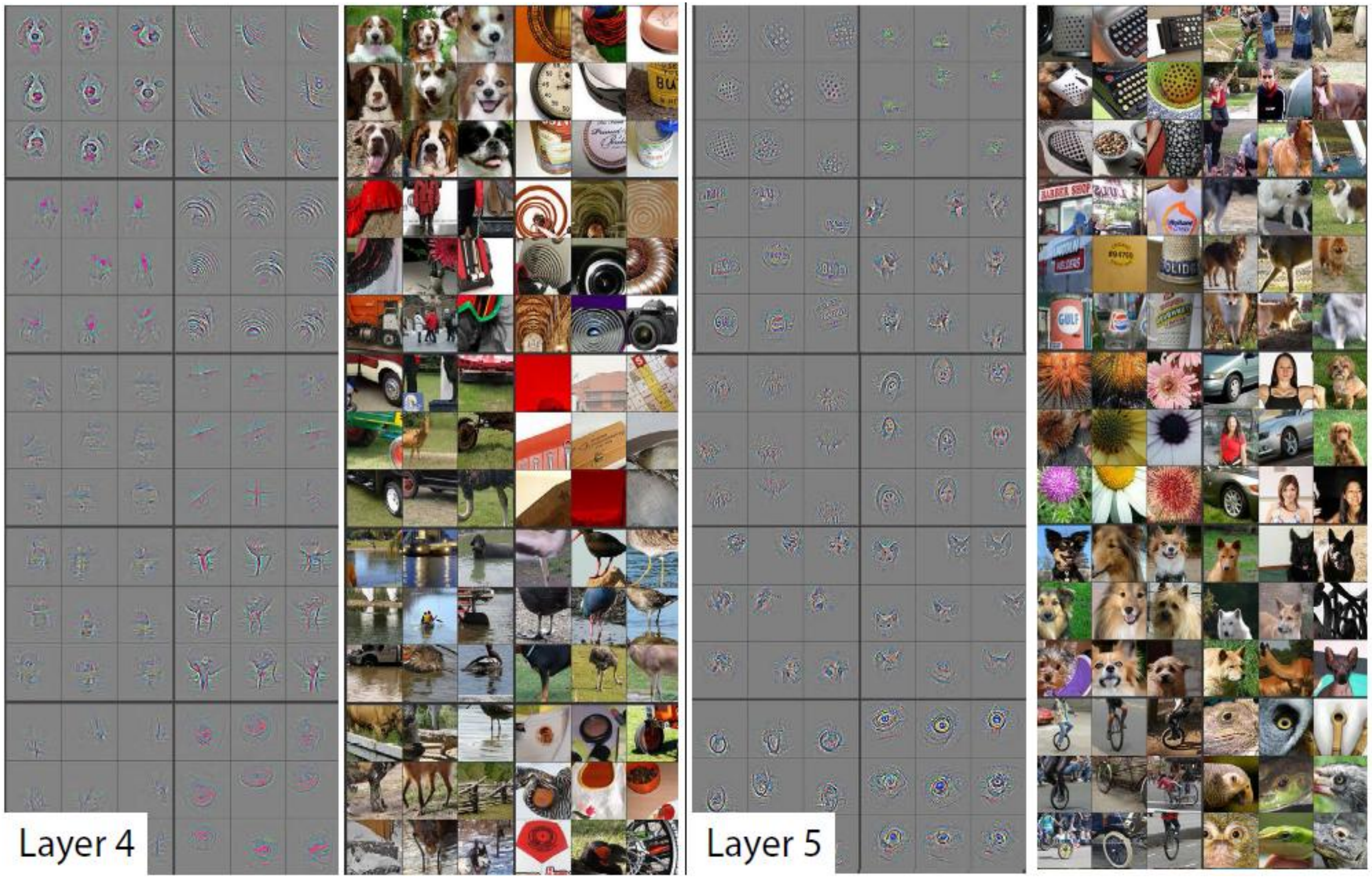
AlexNet Filters- DeconvNet





[images from https://arxiv.org/pdf/1311.2901.pdf]

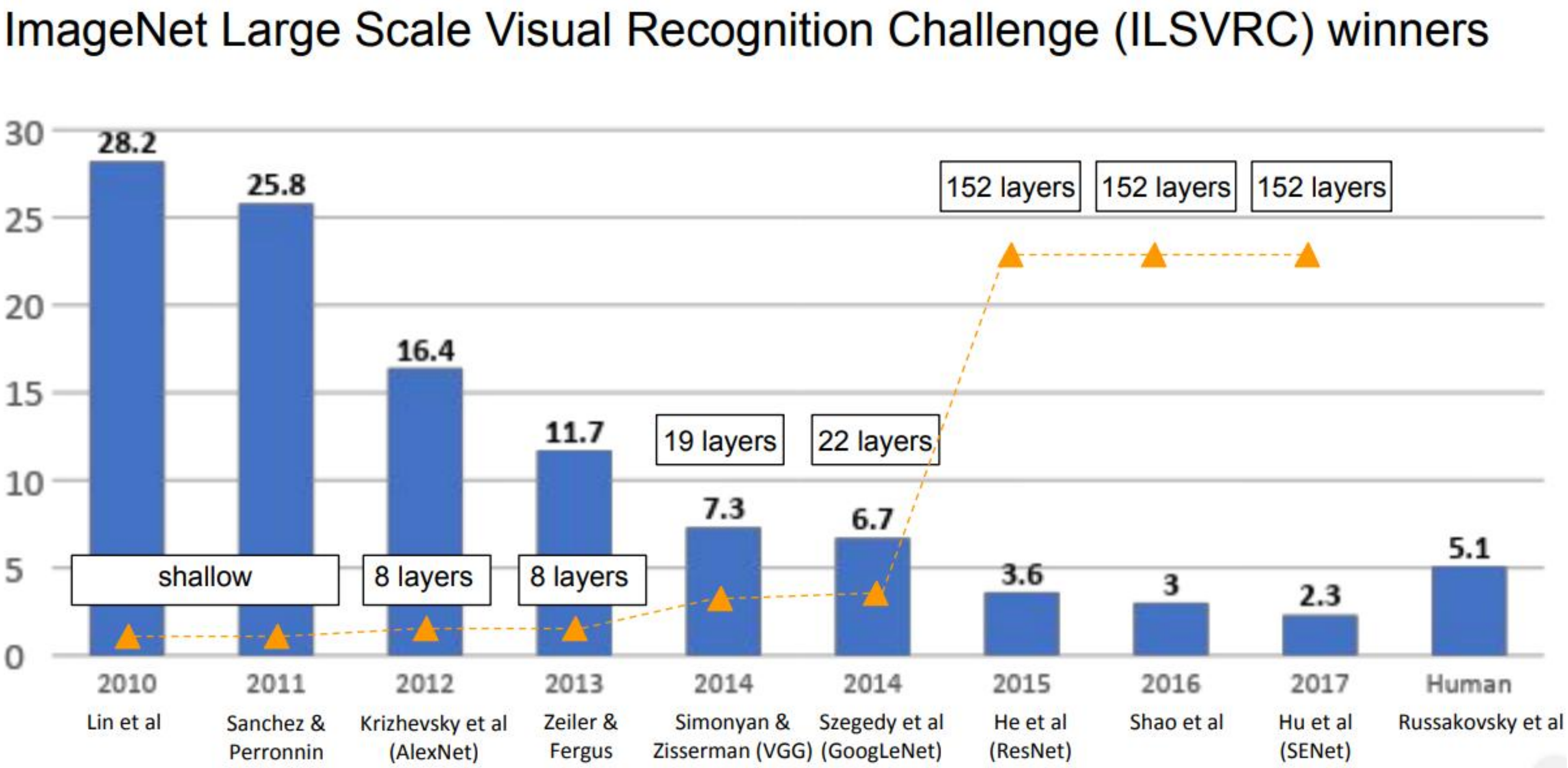
AlexNet Filters- DeconvNet



[images from https://arxiv.org/pdf/1311.2901.pdf]

Beyond AlexNet:
The DCNN storm

ImageNet: the full story



The challenge is now over

Image from
[http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture09.pdf]

VGG Architecture

Several variants

Only 3x3 convolutional filters used
(each with ReLU)

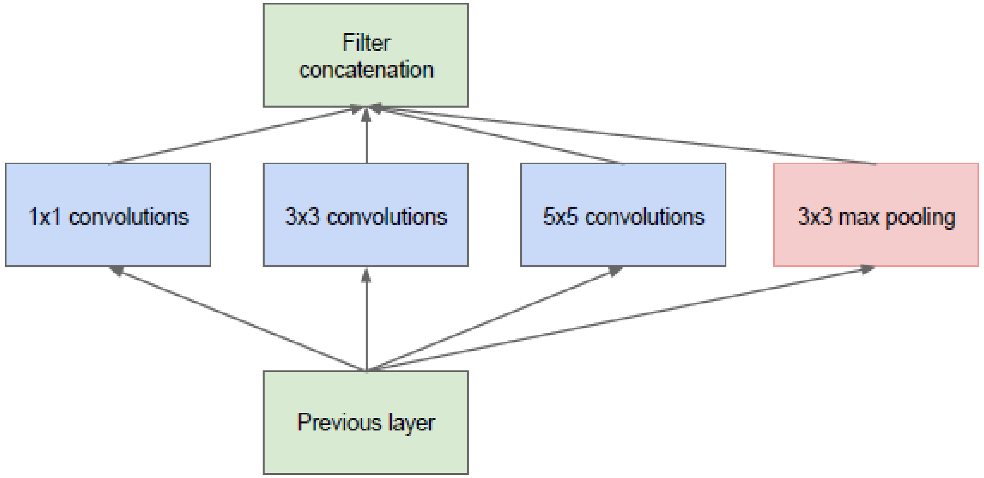
LRN used in only one variant

Image from
[https://arxiv.org/pdf/1409.1556.pdf]

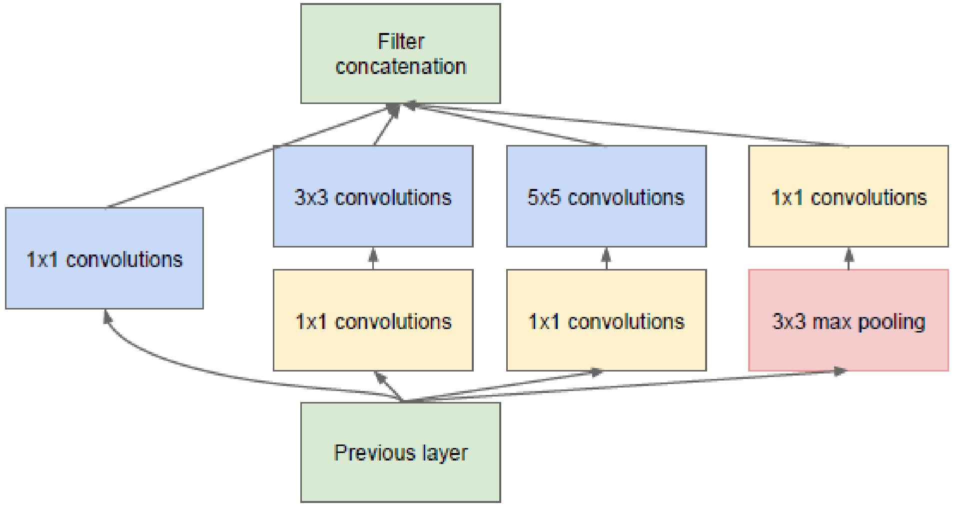
ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Inception Architecture

- Inception modules



(a) Inception module, naïve version

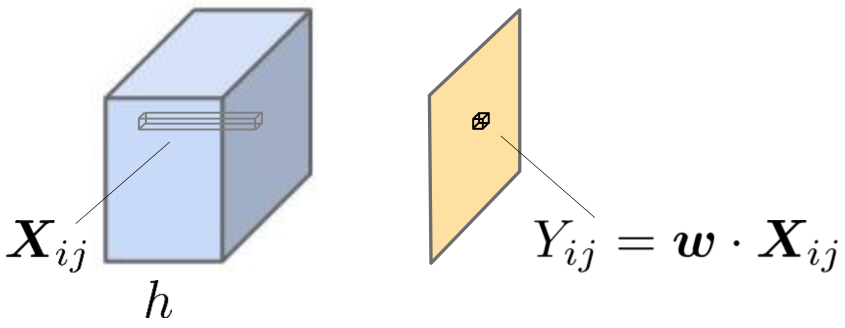


(b) Inception module with dimension reductions

Image from [https://arxiv.org/pdf/1409.4842.pdf]

Inception Architecture

- 1x1 convolution?



(It is a kind of misnomer)
Each filter has dimension $1 \times 1 \times h$
where h is the depth of the set of filter maps

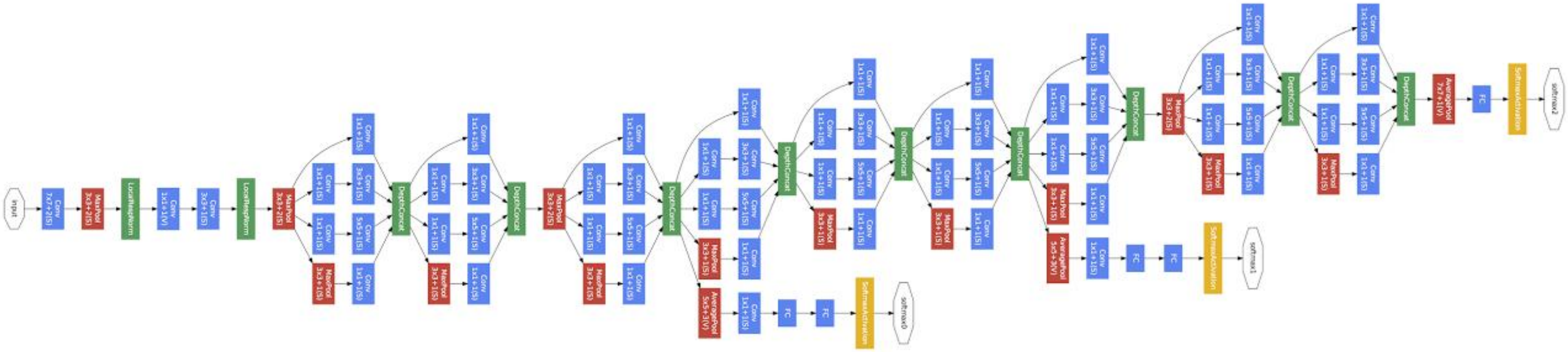
Using d 1x1 convolution filters allows changing depth h into d

Clearly the assumption is $d < h$

It mimics a fully connected layer (across channels)

Inception Architecture

- GoogLeNet architecture

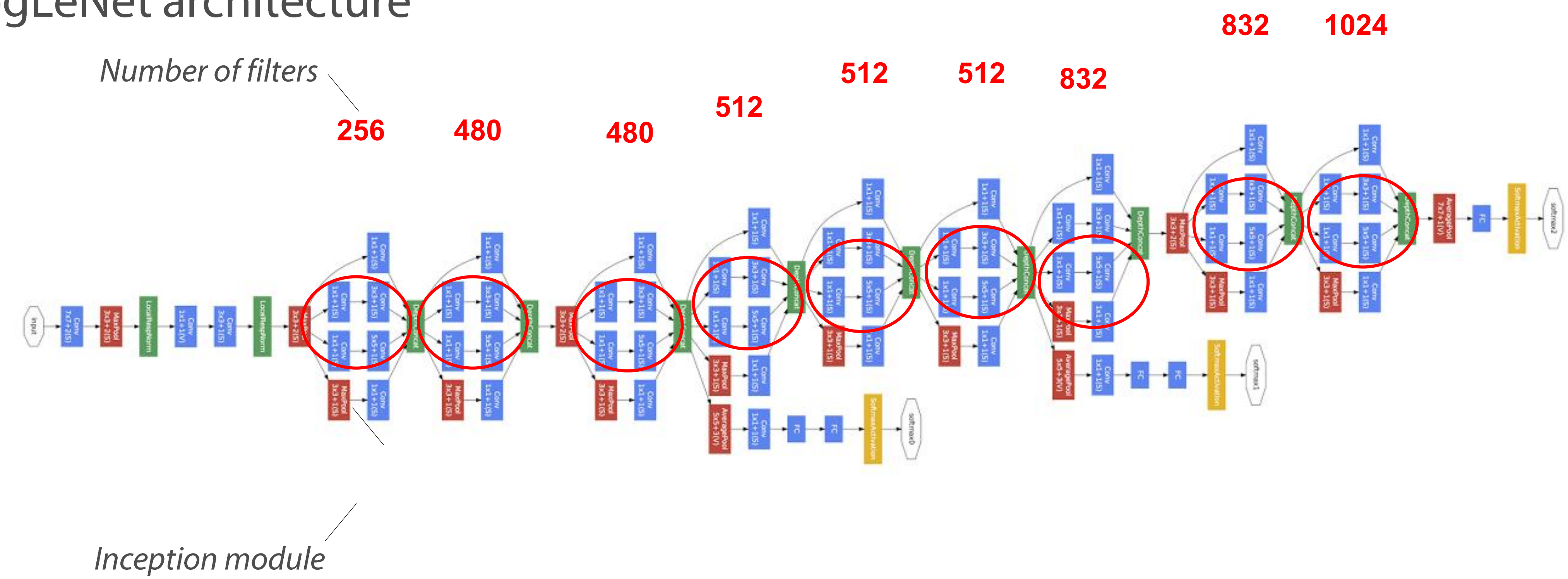


Convolutive
Max Pool
Softmax
Filter Concat

Image from [https://arxiv.org/pdf/1409.4842.pdf]

Inception Architecture

- GoogLeNet architecture

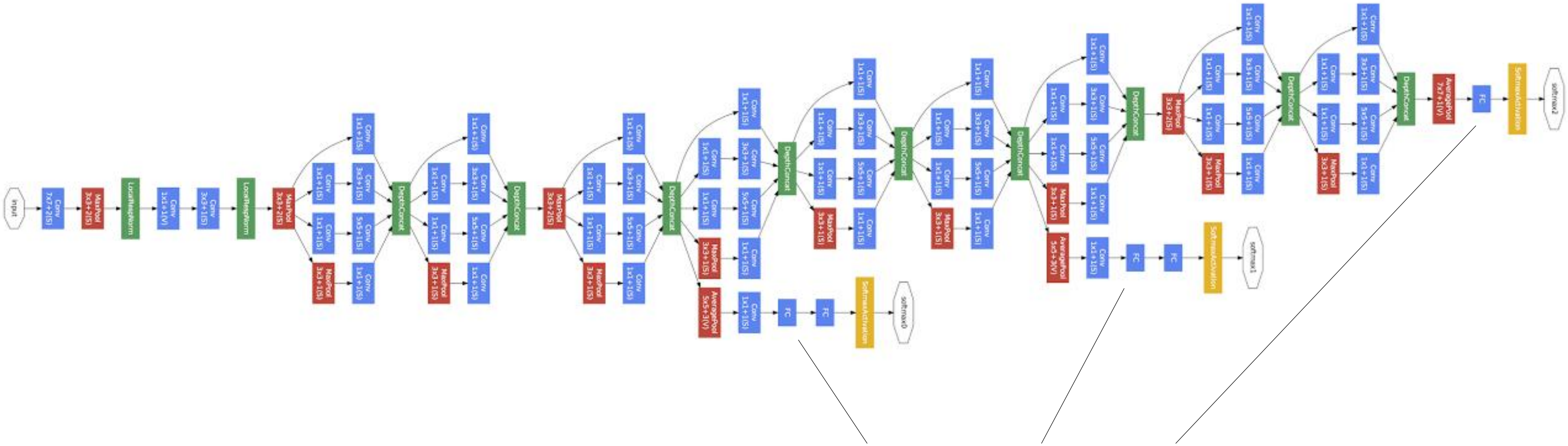


Convolutive
Max Pool
Softmax
Filter Concat

Image from [https://arxiv.org/pdf/1409.4842.pdf]

Inception Architecture

- GoggLeNet architecture



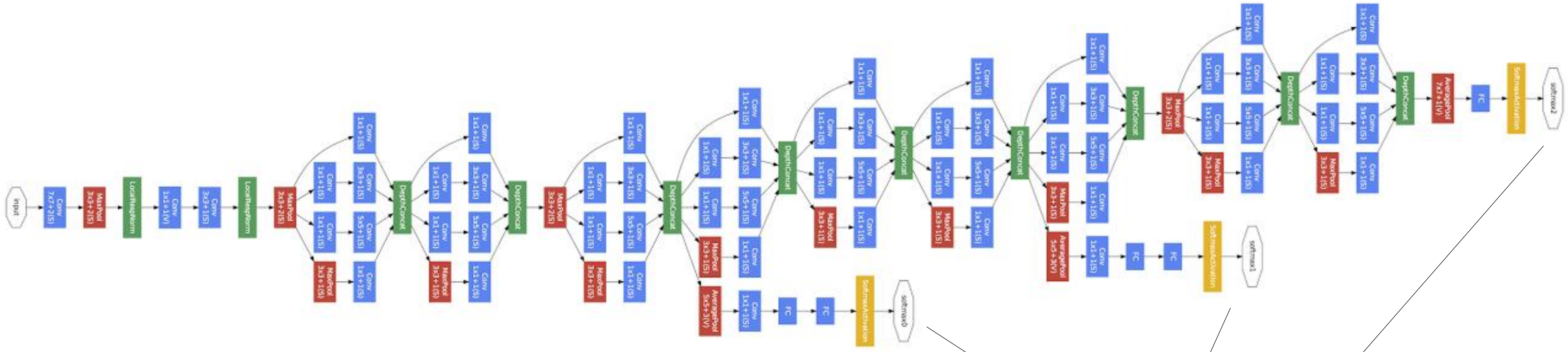
Much smaller FC layers

Convolutive
Max Pool
Softmax
Filter Concat

Image from [https://arxiv.org/pdf/1409.4842.pdf]

Inception Architecture

- GoogLeNet architecture



Three softmax outputs

They are trained to produce
the same output, simultaneously

Convolutive
Max Pool
Softmax
Filter Concat

Image from [https://arxiv.org/pdf/1409.4842.pdf]

ResNet Architecture

- ResNet block

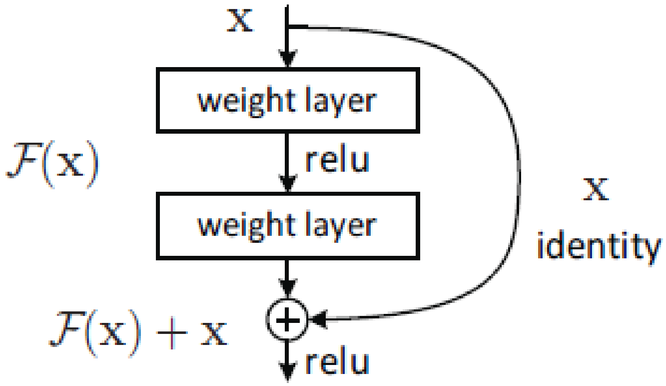


Figure 2. Residual learning: a building block.

Image from [https://arxiv.org/pdf/1512.03385.pdf]

ResNet Architecture

- ResNet architecture

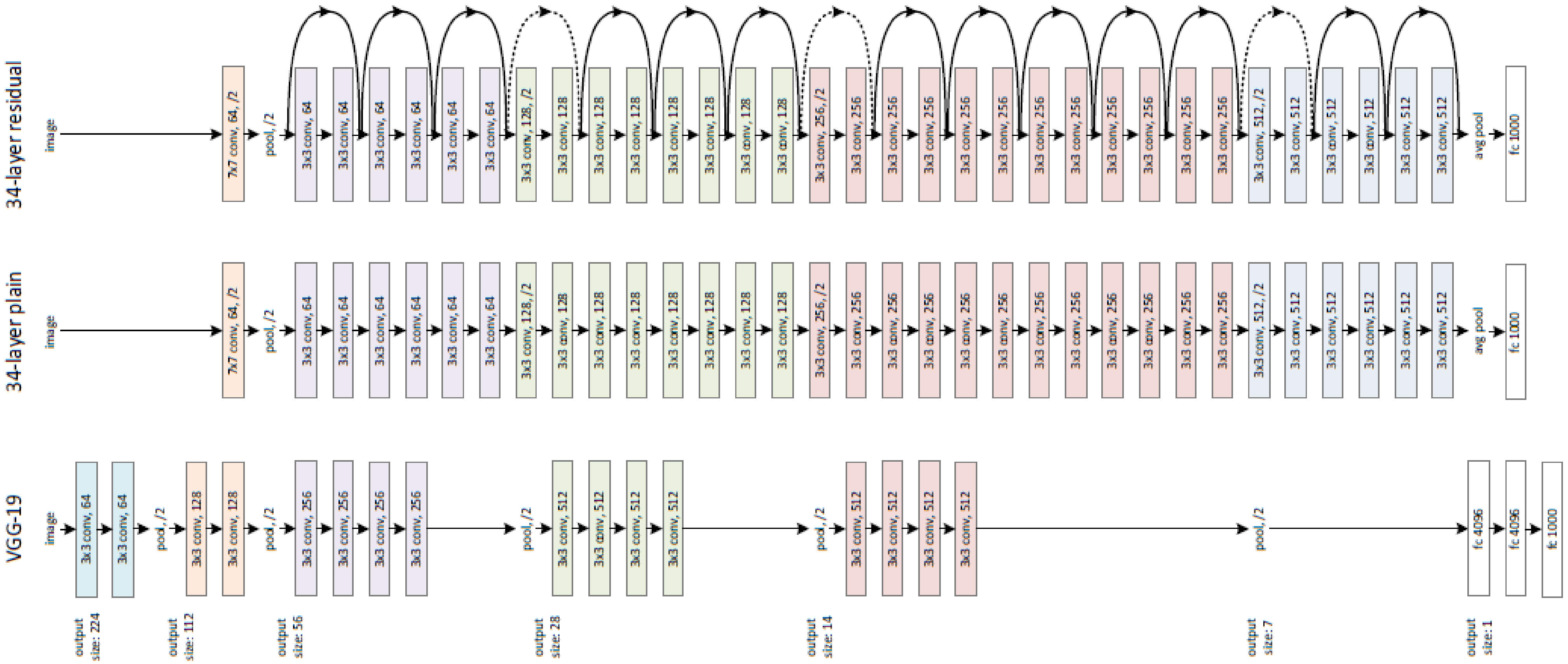
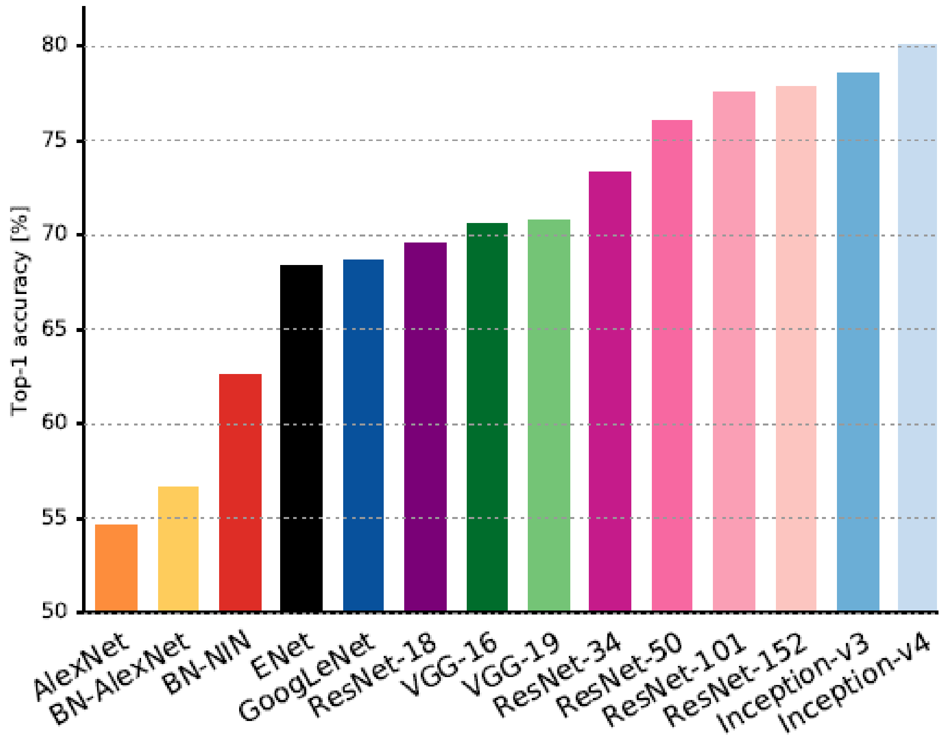


Image from [https://arxiv.org/pdf/1512.03385.pdf]

Comparing Different DCNNs

- Comparative charts at Top-1 accuracy

  i.e. how often the DCNN is right with ImageNet with its top prediction



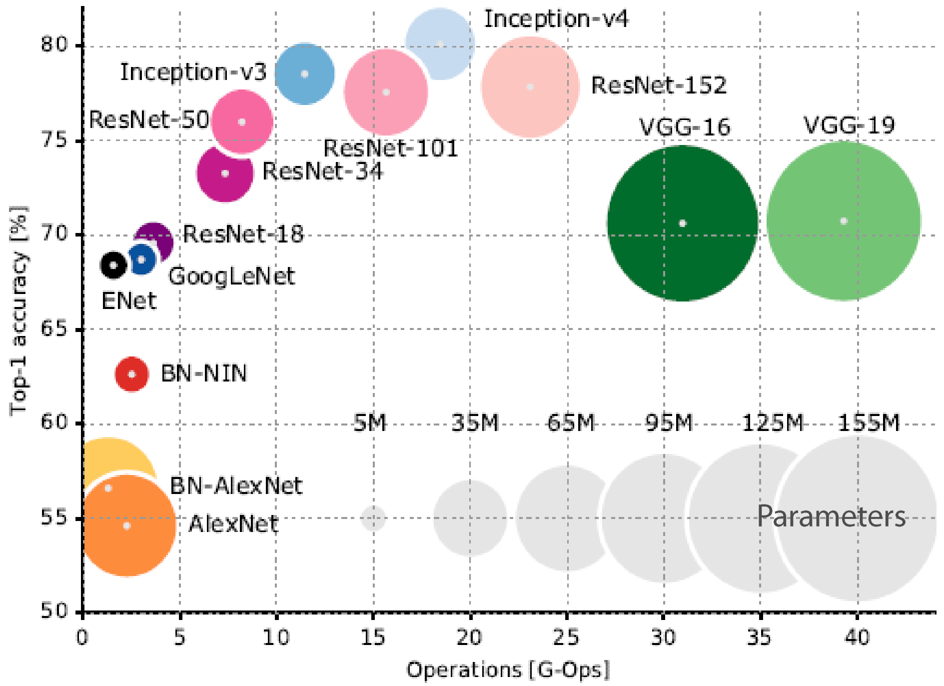


Image from [https://arxiv.org/abs/1605.07678, 2017]

Comparing Different DCNNs

(Same chart, a more recent version)

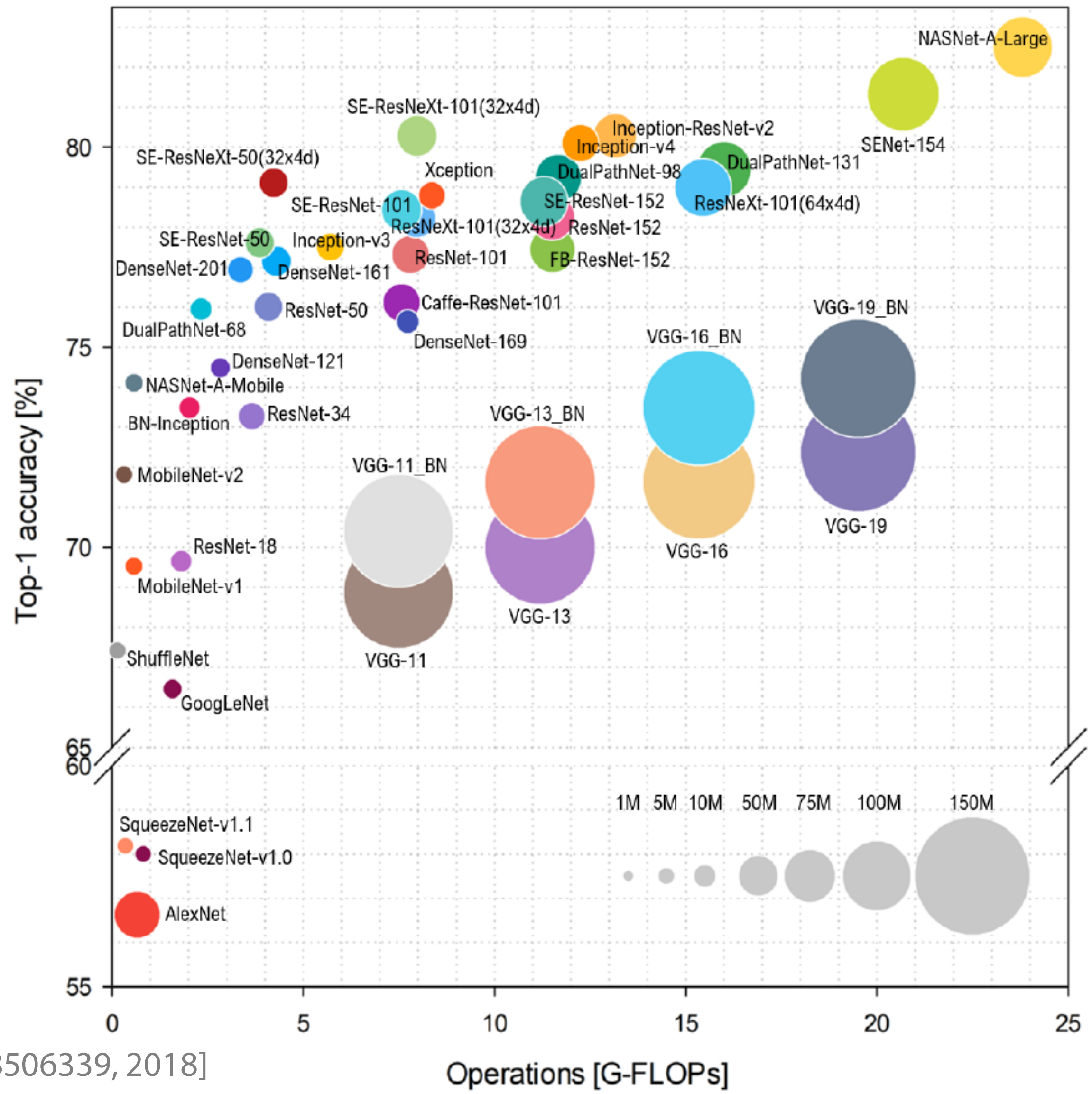


Image from [https://ieeexplore.ieee.org/document/8506339, 2018]

Transfer Learning

Transfer Learning

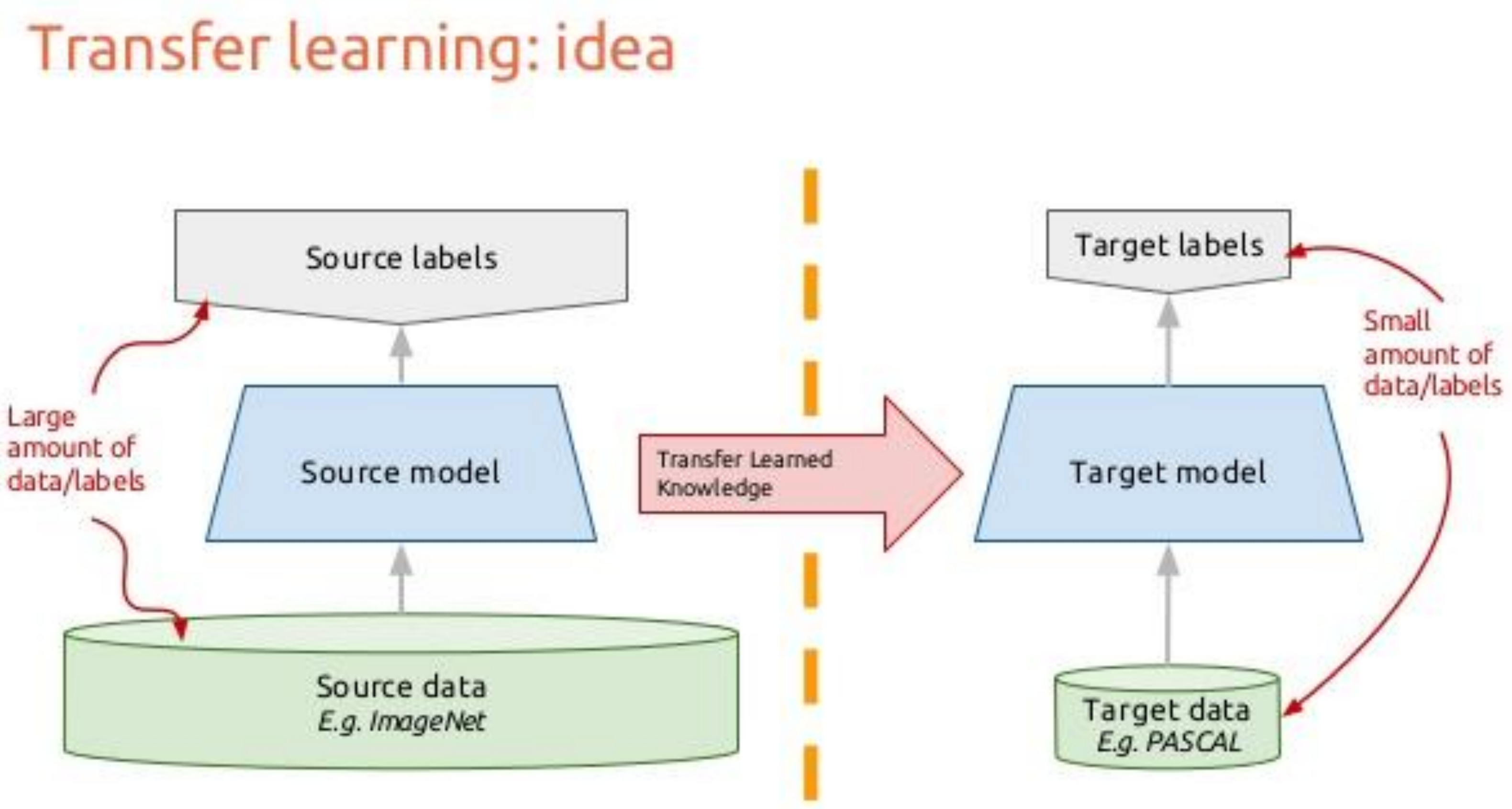


Image from https://www.slideshare.net/xavigiro/transfer-learning-d2l4-insightdcu-machine-learning-workshop-2017

Transfer Learning

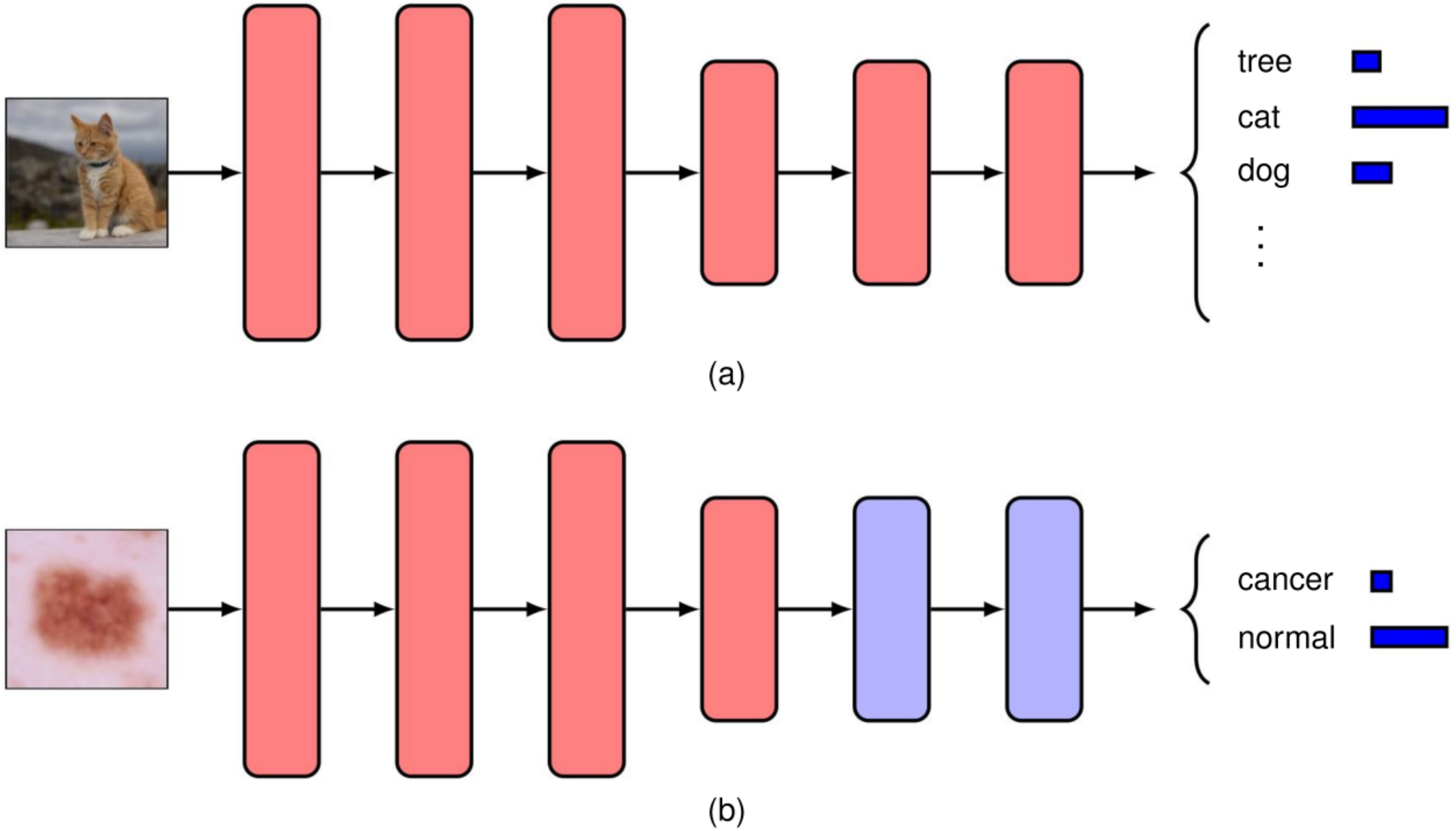


Image from https://www.bishopbook.com/

Do DCNNs Dream of Electric Sheep?

Can DCNNs 'dream'?

sign in | become a supporter | subscribe | search | jobs | dating | more | International edition

theguardian

UK world sport football opinion culture business lifestyle fashion environment tech travel | browse all sections

home › tech

Artificial intelligence (AI)

Yes, androids do dream of electric sheep

Google sets up feedback loop in its image recognition neural network - which looks for patterns in pictures - creating hallucinatory images of animals, buildings and landscapes which veer from beautiful to terrifying

This article is 1 year old

109,591 | 445

Alex Hern

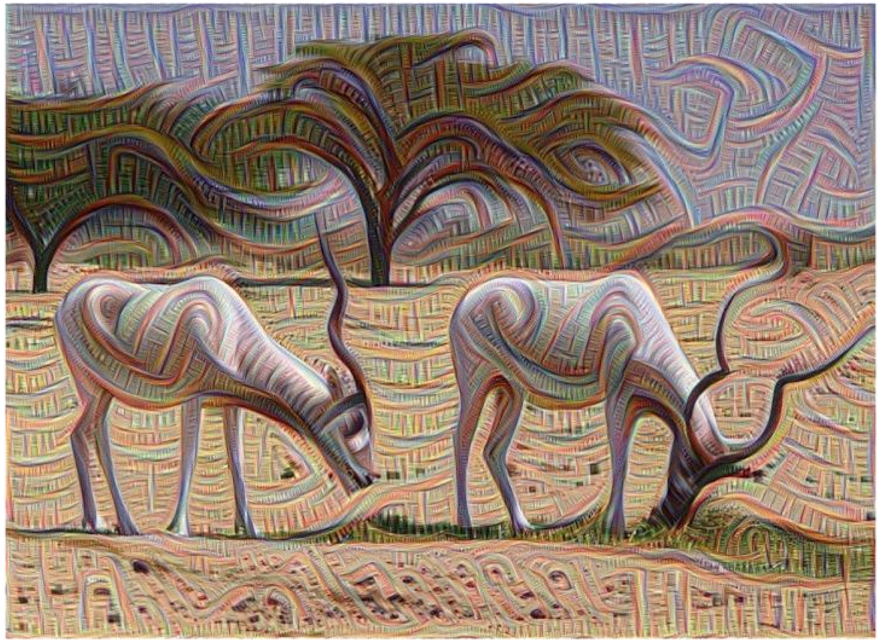
@alexhern

Thursday 18 June 2015 12.57 BST

A hallucinatory filter over a red tree. Spot the animals. Photograph: Google

Can DCNNs 'dream'?

Enhancing lower layers



[images from https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html]

Feature Enhancement

- **Image Space Gradient Descent**

  Define

  $$\boldsymbol{\Phi}_{k,l}(\boldsymbol{I})$$

  as the response of a DCNN at a layer k, filter l to an image $\boldsymbol{I}$

  Given a specific image $\hat{\boldsymbol{I}}$, we define the loss function

  $$L(\hat{\boldsymbol{I}}, \boldsymbol{I}) := \|\gamma\, \boldsymbol{\Phi}_{k,l}(\hat{\boldsymbol{I}}) - \boldsymbol{\Phi}_{k,l}(\boldsymbol{I})\|^2$$

  Amplification factor

  The optimization problem

  $$\boldsymbol{I}^* := \mathrm{argmin}_{\boldsymbol{I}} \left(L(\hat{\boldsymbol{I}}, \boldsymbol{I}) + \lambda \|\boldsymbol{I}\|^2 \right)$$

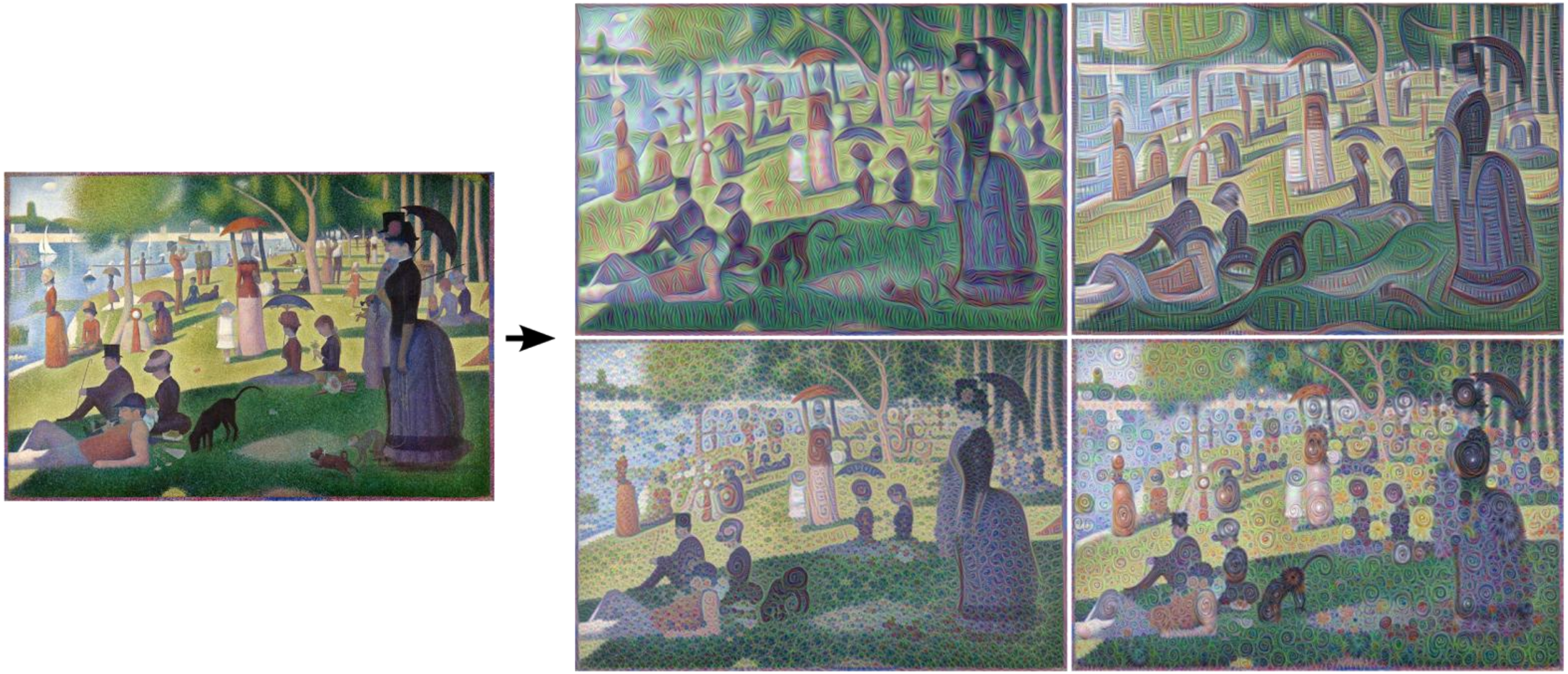
  is solved via gradient descent by computing

  $$\frac{\partial}{\partial \boldsymbol{I}} \left(L(\hat{\boldsymbol{I}}, \boldsymbol{I}) + \lambda \|\boldsymbol{I}\|^2 \right)$$

  and starting from $\boldsymbol{I}^{(0)} = \hat{\boldsymbol{I}}$

Can DCNNs 'dream'?

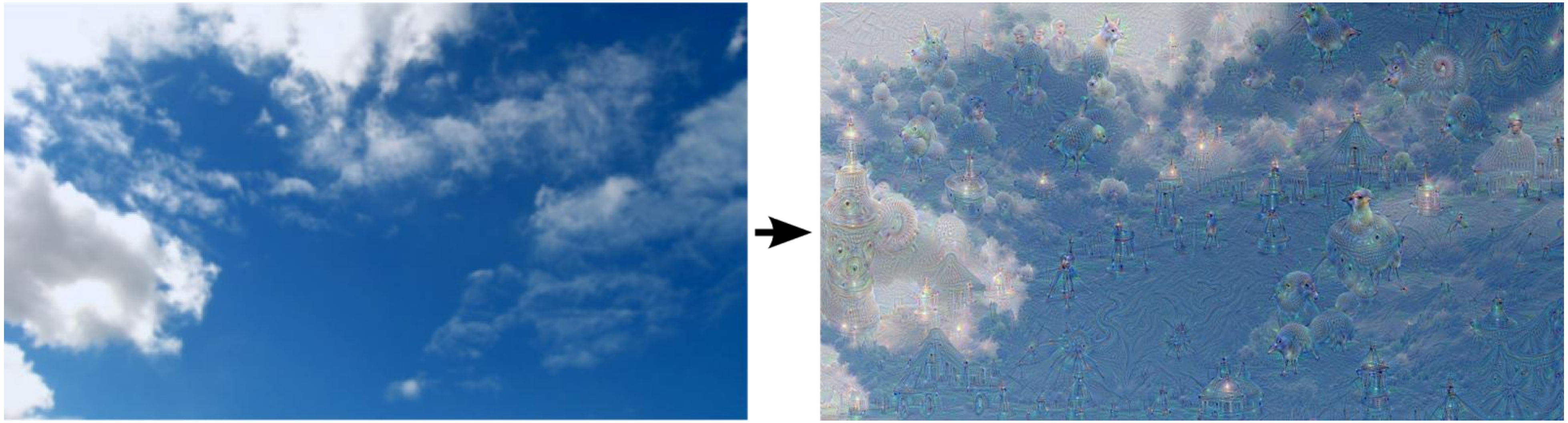
Enhancing lower layers



[images from https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html]

Can DCNNs 'dream'?

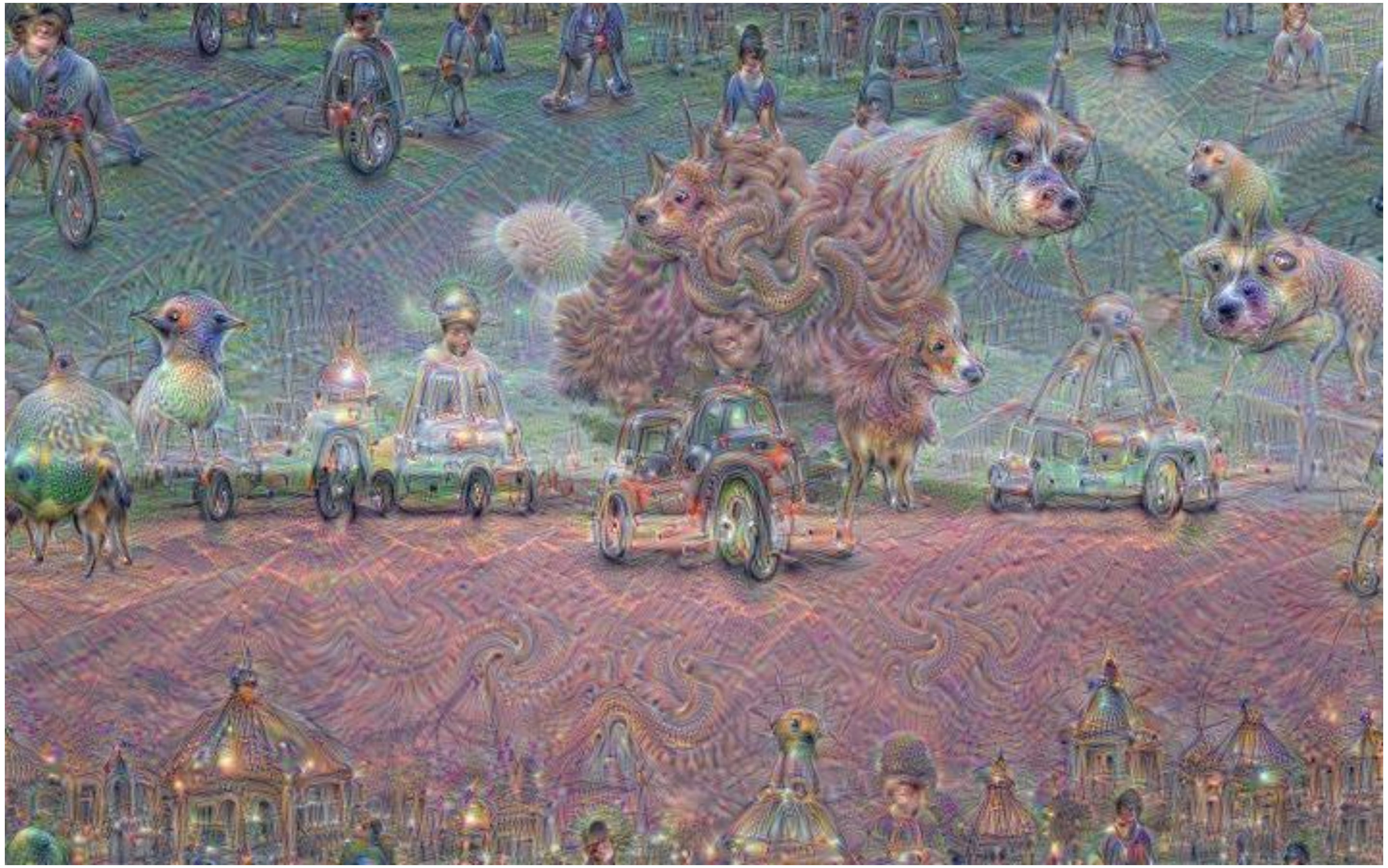
Enhancing upper layers



[images from https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html]

Can DCNNs 'dream'?

Letting the DCNN go on its own



[images from https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html]

Can DCNNs 'dream'?

Letting the DCNN go on its own

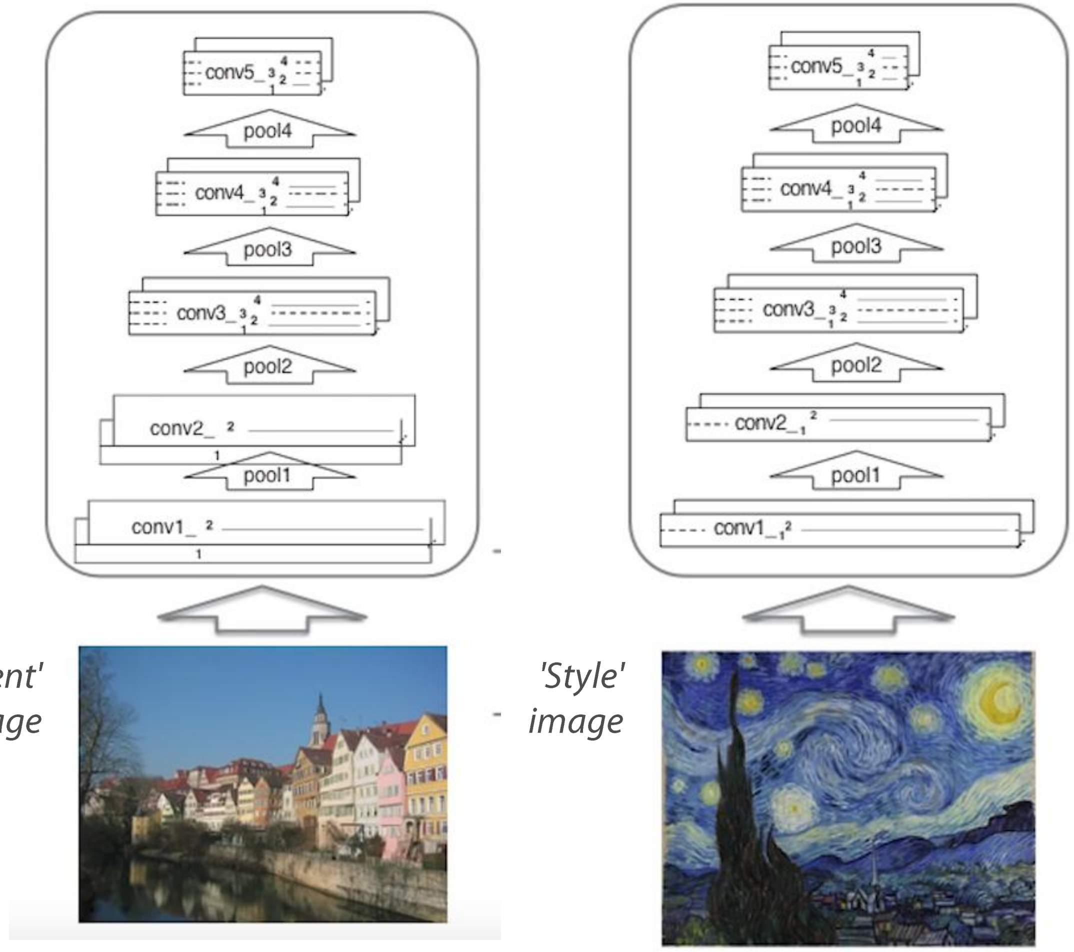
[images from https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html]

The Power of Abstraction
(in layers)

The Power of Abstraction

- ***Different Layers of a Deep Convolutional Neural Network***

  What kind of information does each layer 'store'?



'Content' image

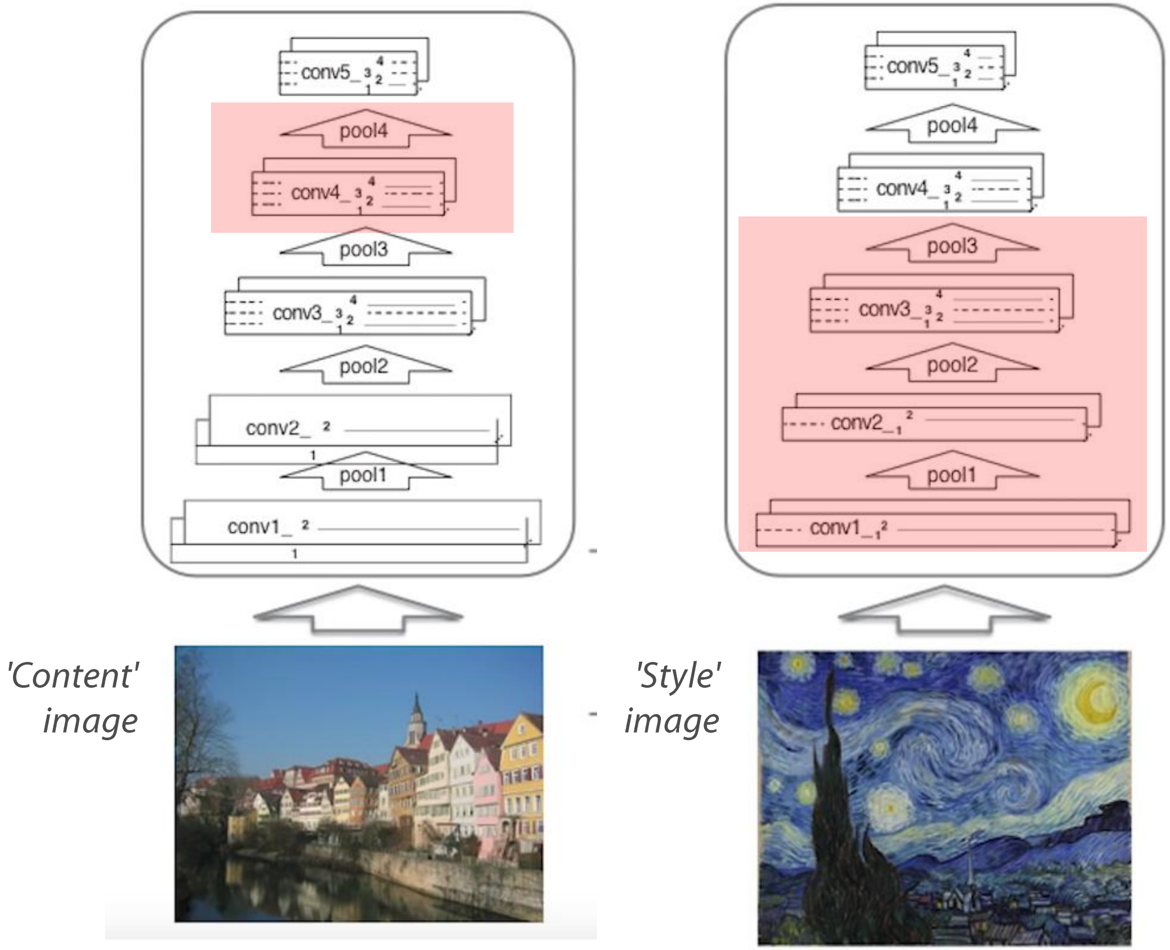
'Style' image

[images from https://www.youtube.com/watch?v=N14_w2RG1A8]

The Power of Abstraction

- ***Different Layers of a Deep Convolutional Neural Network***

  What kind of information does each layer 'store'?



Create a new image
by combining more
of the 'Content' top layer
and more of 'Style' low layers

[images from https://www.youtube.com/watch?v=N14_w2RG1A8]

Mixing Two Images

- **Image Space Gradient Descent**

  Define

  $$\boldsymbol{\Phi}_{k,l}(\boldsymbol{I})$$

  as the response of a DCNN at a layer k, filter l to an image $\boldsymbol{I}$

  Given a specific image $\hat{\boldsymbol{I}}_1$ and $\hat{\boldsymbol{I}}_2$, we define the loss function

  $$L(\hat{\boldsymbol{I}}, \boldsymbol{I}) := \sum_{k,l} \|\boldsymbol{M}_{k,l}(\boldsymbol{\Phi}_{k,l}(\hat{\boldsymbol{I}}_2), \boldsymbol{\Phi}_{k,l}(\hat{\boldsymbol{I}}_1)) - \boldsymbol{\Phi}_{k,l}(\boldsymbol{I})\|^2$$

  Weighted Merge Function

  The optimization problem

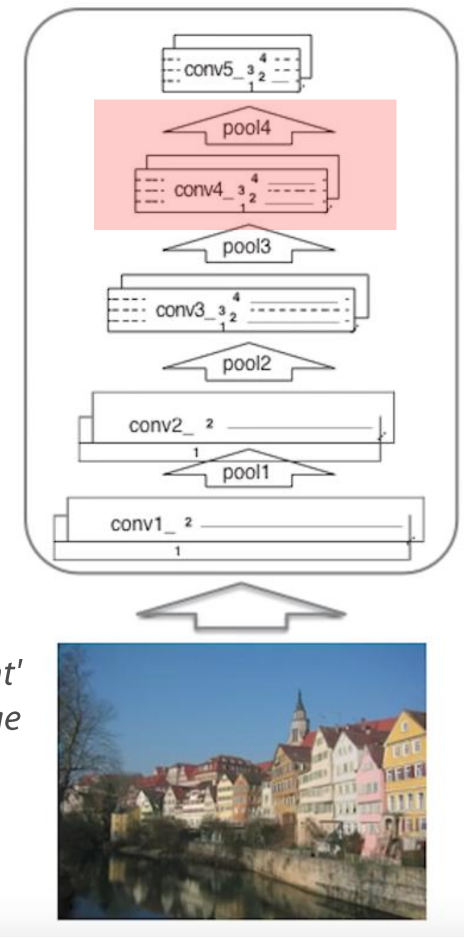
  $$\boldsymbol{I}^* := \operatorname{argmin}_{\boldsymbol{I}} \left(L(\hat{\boldsymbol{I}}, \boldsymbol{I}) + \lambda \|\boldsymbol{I}\|^2 \right)$$

  is solved via gradient descent starting from $\boldsymbol{I}^{(0)} = \hat{\boldsymbol{I}}_1$

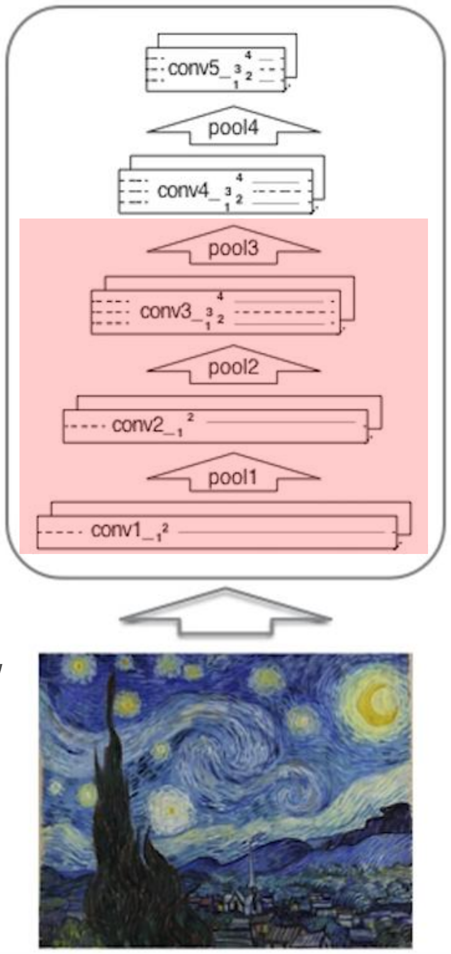
The Power of Abstraction

- ***Different Layers of a Deep Convolutional Neural Network***

  What kind of information does each layer 'store'?



'Content' image



'Style' image

Create a new image by combining more of the 'Content' top layer and more of 'Style' low layers

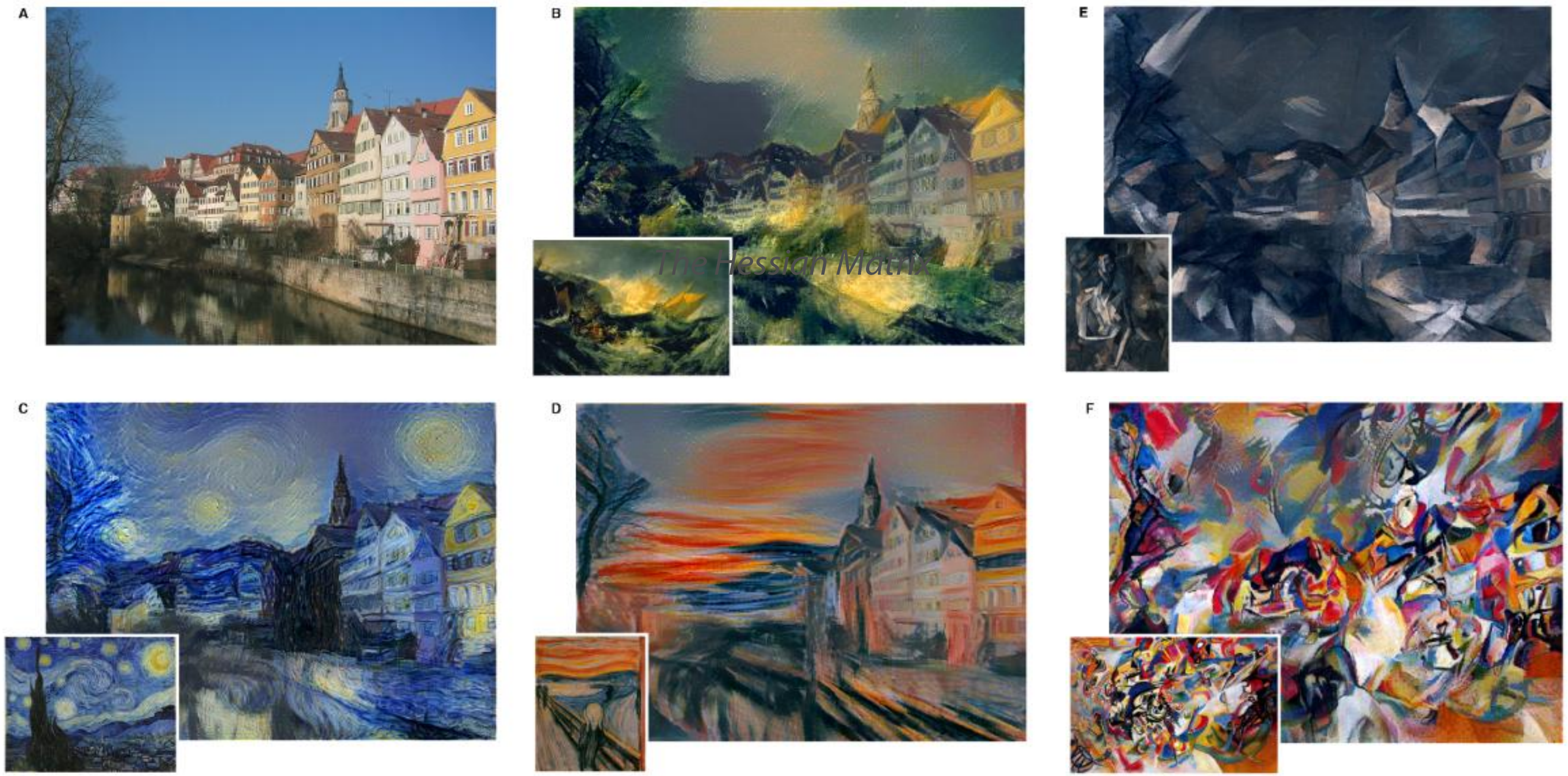
This is the result

[images from https://www.youtube.com/watch?v=N14_w2RG1A8]

The Power of Abstraction

- ***Different Layers of a Deep Convolutional Neural Network***

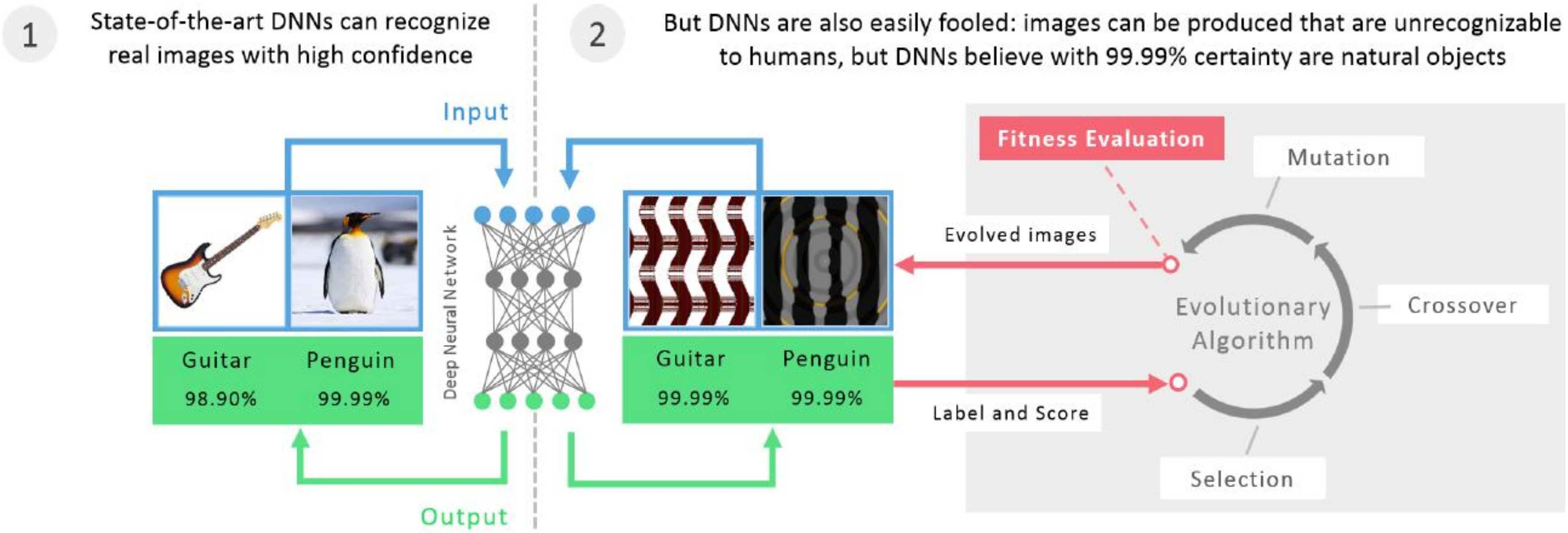
  Further examples:



[images from https://arxiv.org/pdf/1508.06576.pdf]

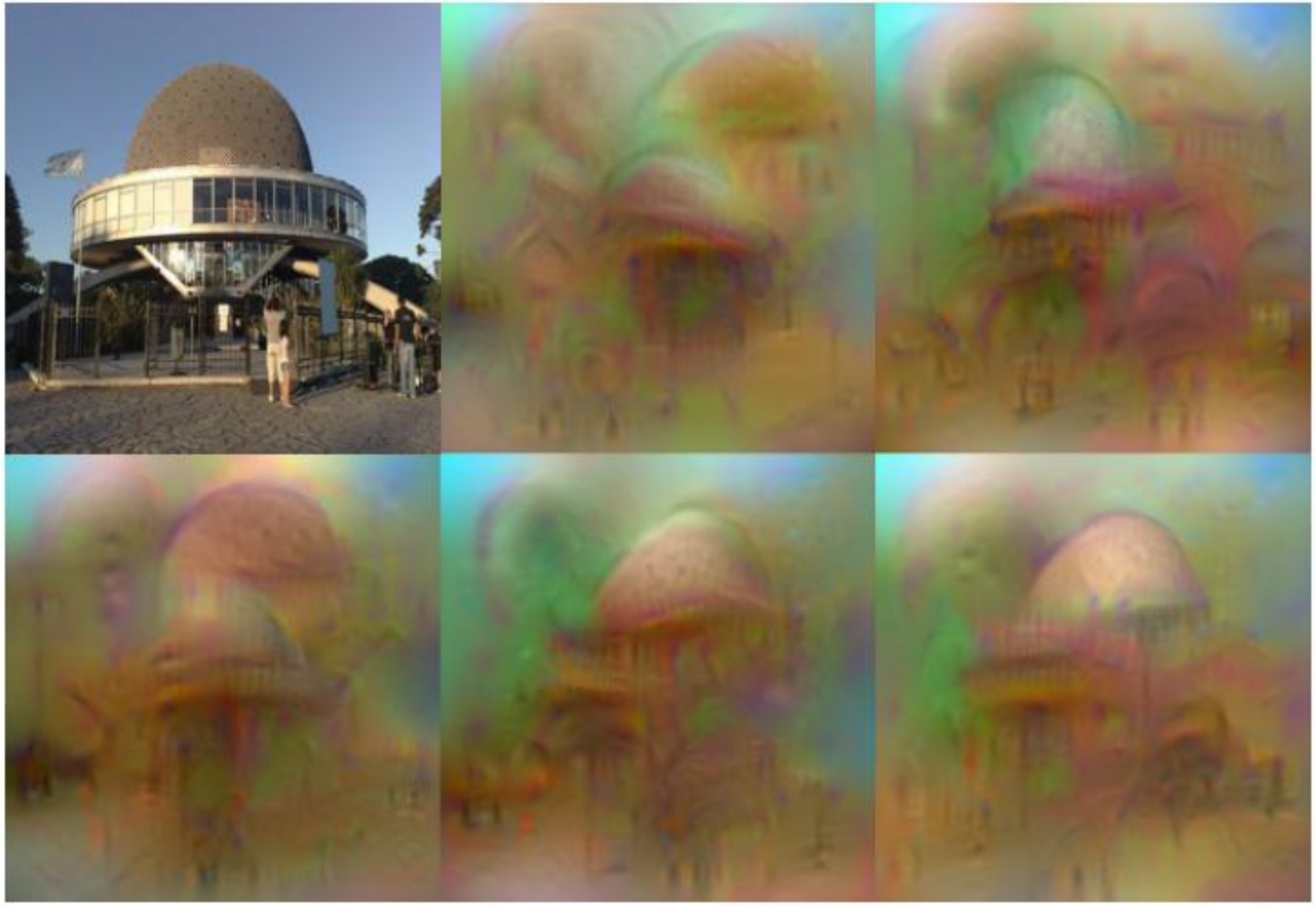
Human-like Vision?
No way!

A DCNN can be fooled...



[images from https://arxiv.org/pdf/1412.1897v4.pdf]

Reconstructing Images from Feature Maps



[images from https://arxiv.org/pdf/1412.0035v1.pdf]

Reconstructing Images from Feature Maps

- **Image Space Gradient Descent**

Define

$$\mathbf{\Phi}_{k,l}(\boldsymbol{I})$$

as the response of a DCNN at a layer k, filter l to an image $\boldsymbol{I}$

Given a specific image $\hat{\boldsymbol{I}}$, we define the loss function

$$L(\hat{\boldsymbol{I}}, \boldsymbol{I}) := \|\mathbf{\Phi}_{k,l}(\hat{\boldsymbol{I}}) - \mathbf{\Phi}_{k,l}(\boldsymbol{I})\|^2$$

and the optimization problem

$$\boldsymbol{I}^* := \mathrm{argmin}_{\boldsymbol{I}} \left(L(\hat{\boldsymbol{I}}, \boldsymbol{I}) + \rho P(\boldsymbol{I}) + \lambda \|\boldsymbol{I}\|^2 \right)$$

L2 Regularization

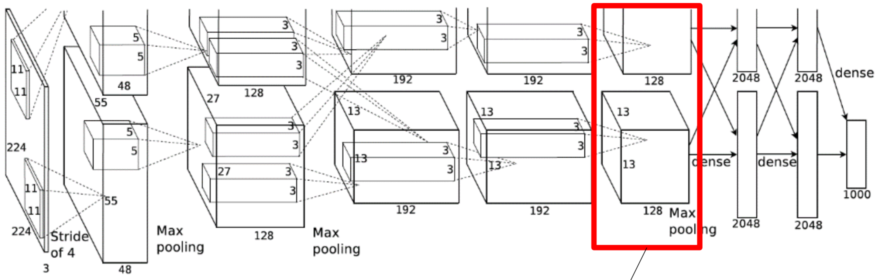
'Statistical Realism'

To solve this, we can compute

$$\frac{\partial}{\partial \boldsymbol{I}} \left(L(\hat{\boldsymbol{I}}, \boldsymbol{I}) + \rho P(\boldsymbol{I}) + \lambda \|\boldsymbol{I}\|^2 \right)$$

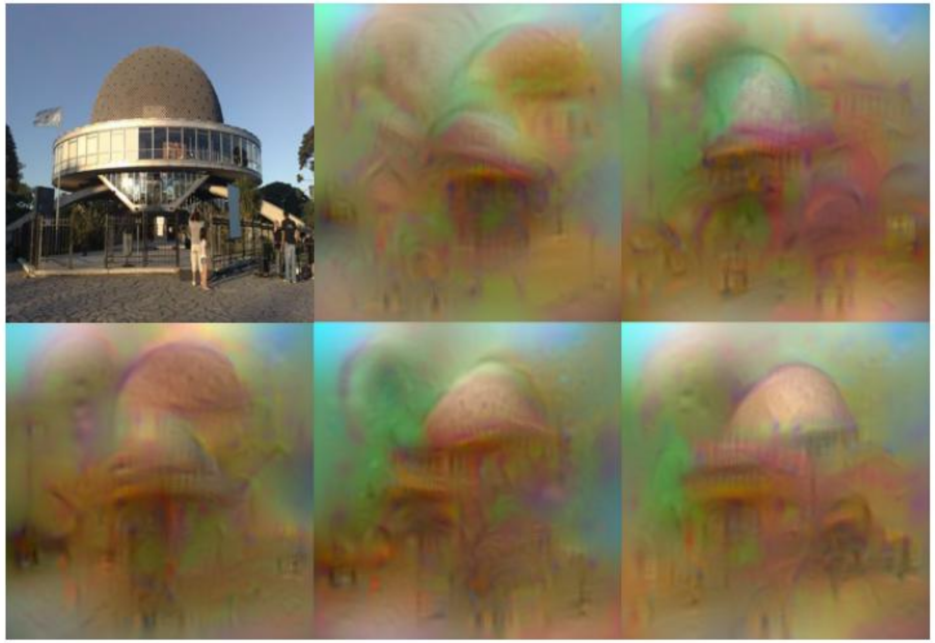
and apply a gradient descent procedure, starting from a random image $\boldsymbol{I}^{(0)}$

Reconstructing Images from Feature Maps



$\Phi_{k,l}(\hat{\boldsymbol{I}})$ is taken here

This is $\hat{\boldsymbol{I}}$



The remaining five images were generated using image space gradient descent with different initial images $\boldsymbol{I}^{(0)}$

[images from https://arxiv.org/pdf/1412.0035v1.pdf]

Just add some little noise ...

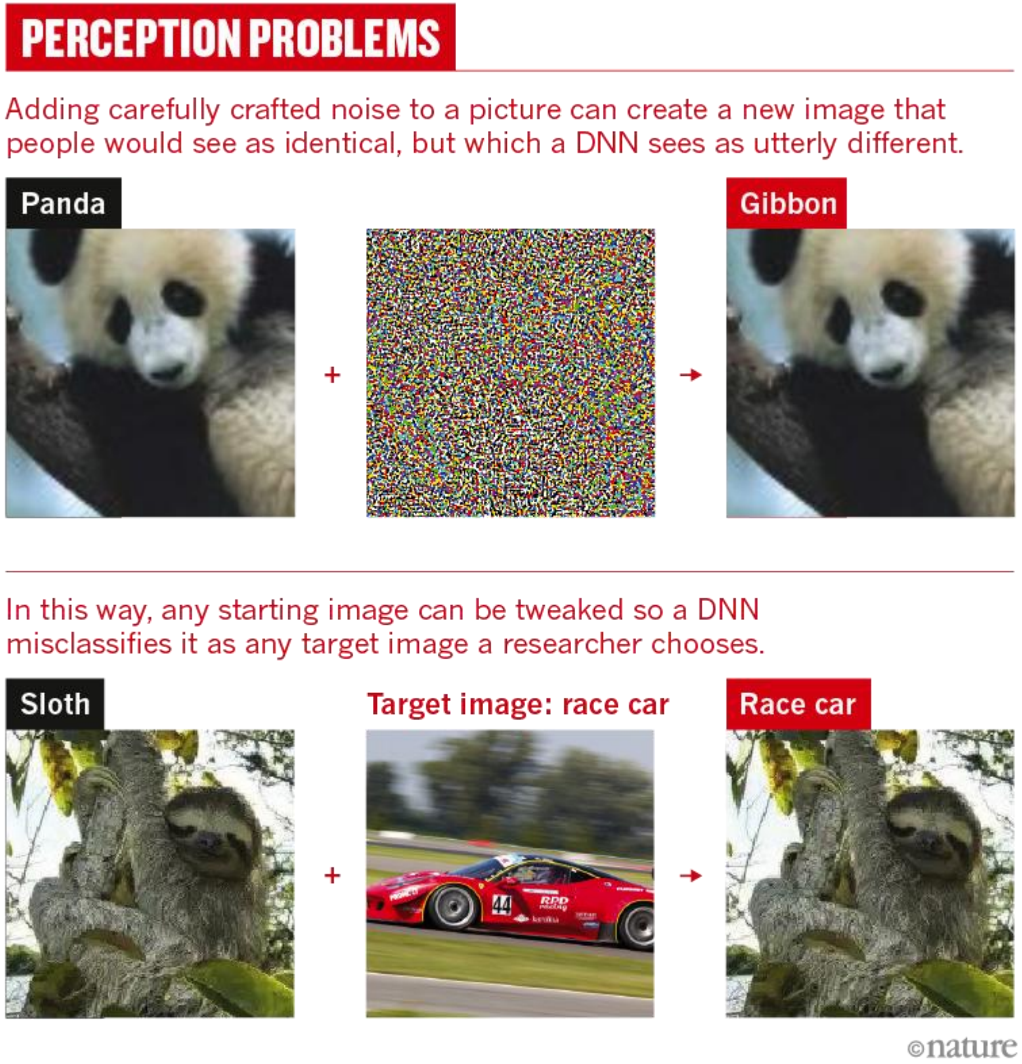
nature

Subscribe

NEWS FEATURE · 09 OCTOBER 2019

Why deep-learning AIs are so easy to fool

Artificial-intelligence researchers are trying to fix the flaws of neural networks.



[images from https://www.nature.com/articles/d41586-019-03013-5