Deep learning hands-on
Lecture 3 Convolutional neural networks

Ole Winther

Dept for Applied Mathematics and Computer Science
Technical University of Denmark (DTU)

August 17, 2018

Objectives of CNN lecture

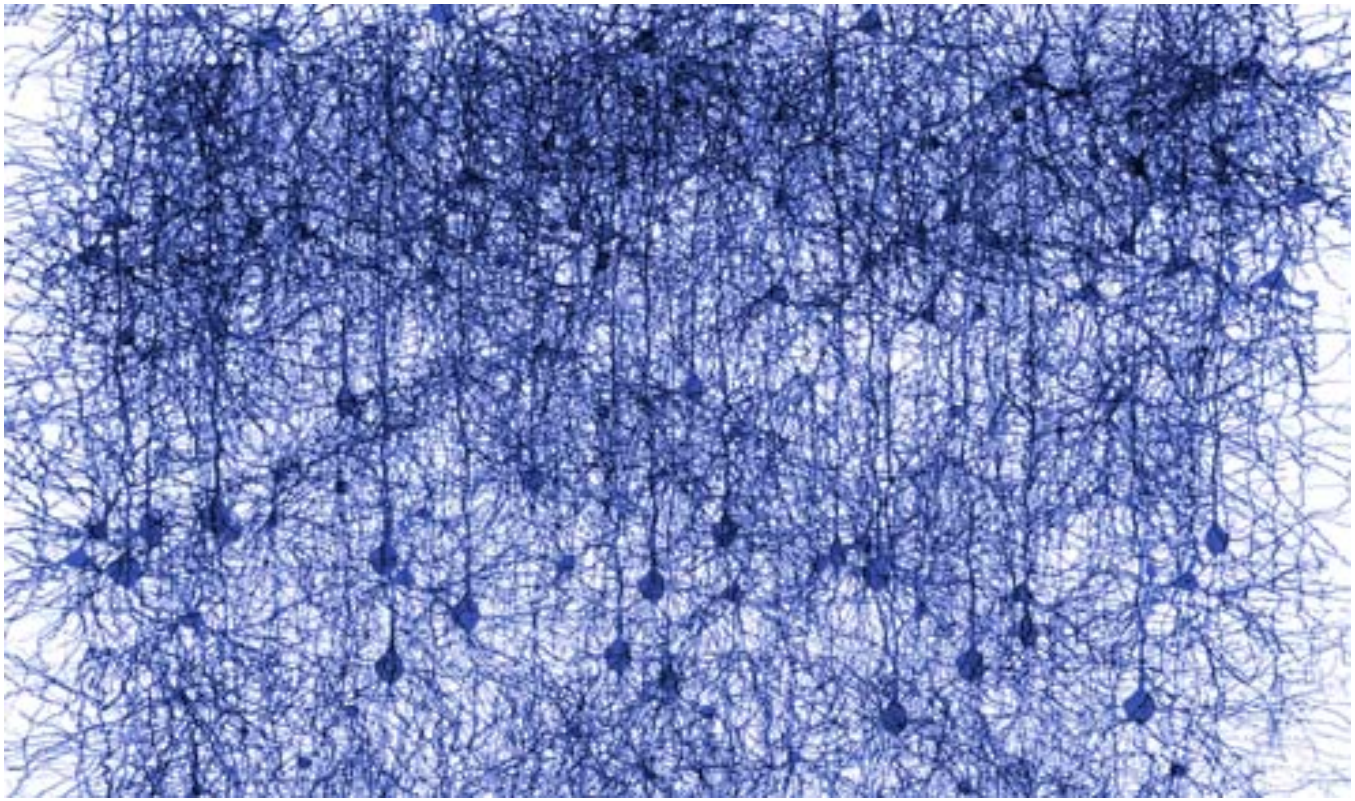
- Convolutional neural networks and
- what they can be used for.
- Example from industrial PhD w Siemens Wind Power

Part 1:

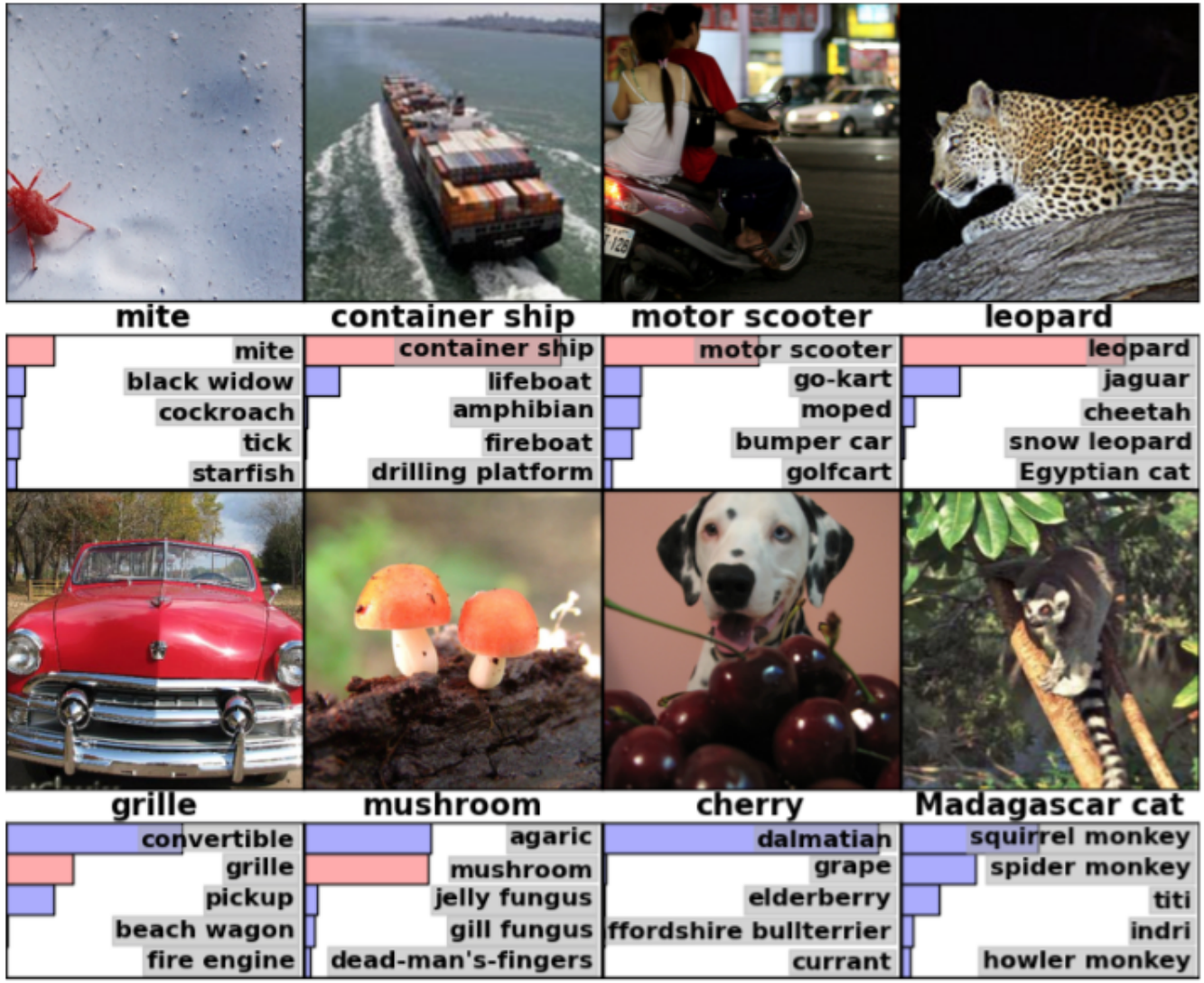
Convolutional NNs
Definition and uses

Neural networks (NNs)

- Feedforward neural networks (FFNNs)
- Convolutional neural networks (CNNs)
- Recurrent Neural Networks (RNNs)
- Auto-encoders (AE)

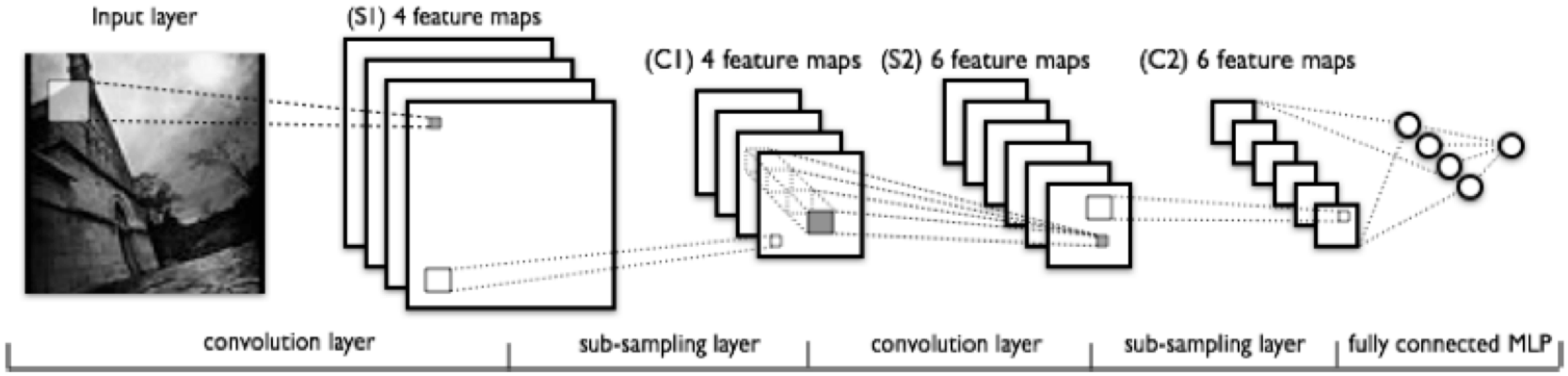


ImageNet - image classification



- 1.000 different classes - including many types of dogs!
- 1.000.000 training images

Convolutional neural networks



$$\begin{bmatrix} 10 & 0 & -10 \\ 0 & 0 & 0 \\ -10 & 0 & 10 \end{bmatrix}$$

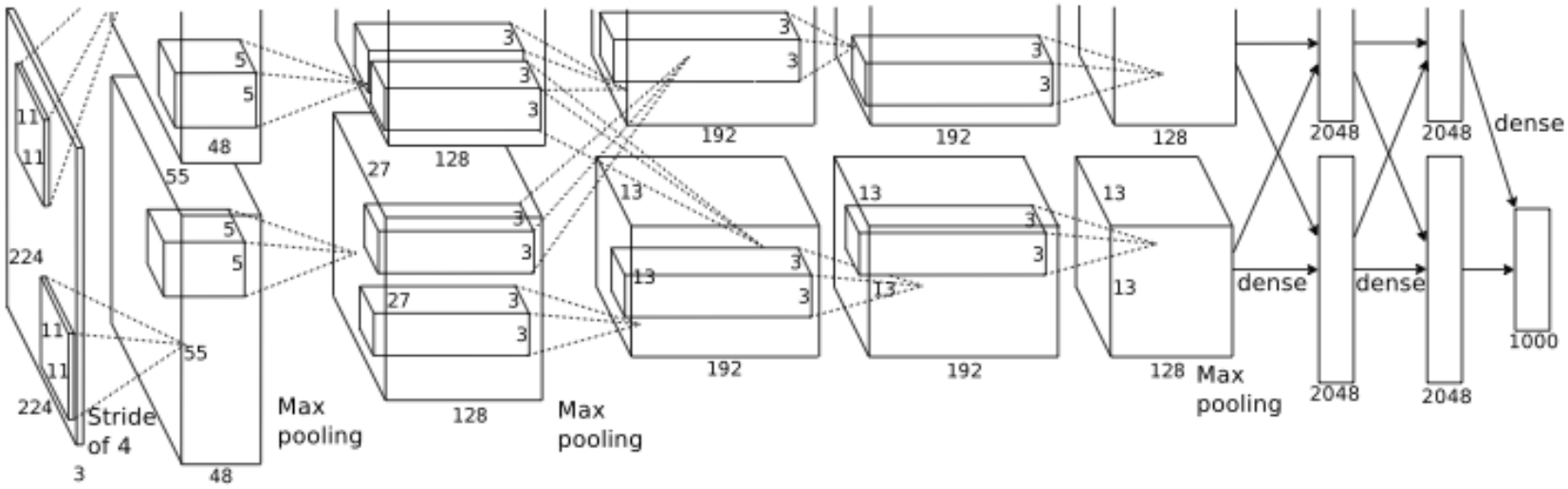
Feature engineering vs engineered models

ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

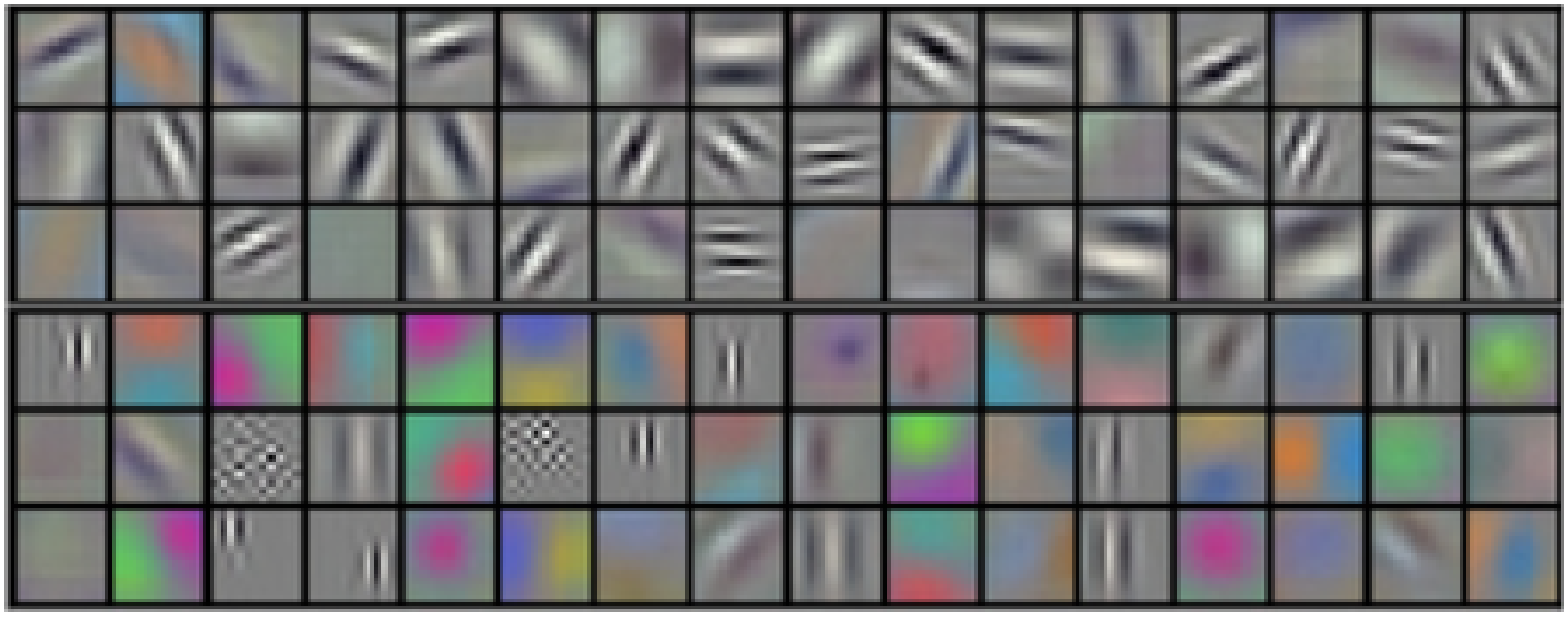
Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca



www.cs.toronto.edu/~fritz/absps/imagenet.pdf

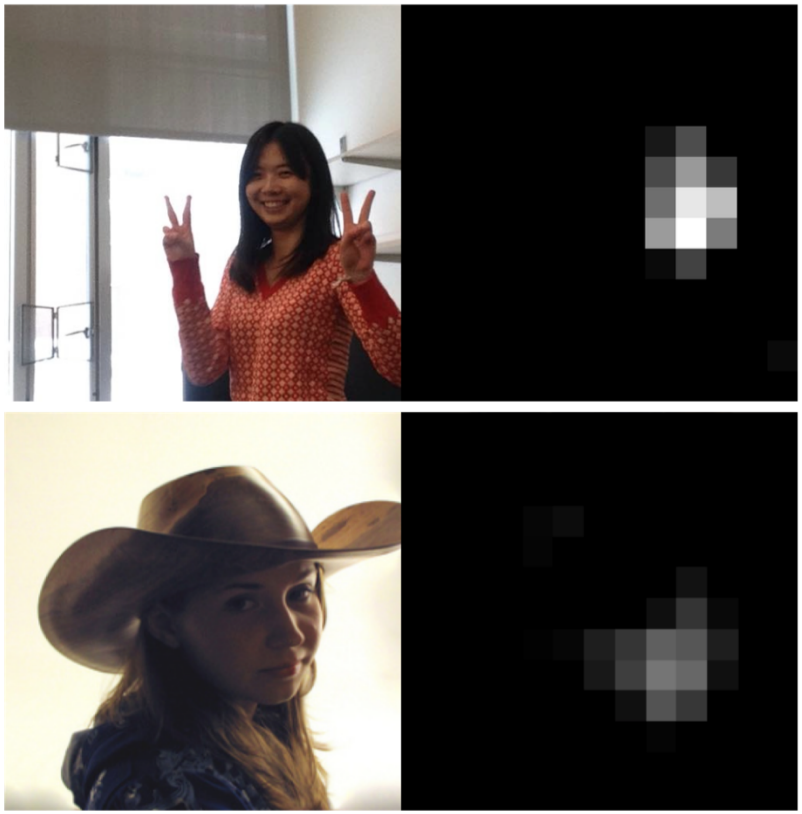
Learned filters in first layer



- Seminal paper:
- Krizhevsky et al, ImageNet Classification with Deep Convolutional Neural Networks
- `www.cs.toronto.edu/~fritz/absps/imagenet.pdf`

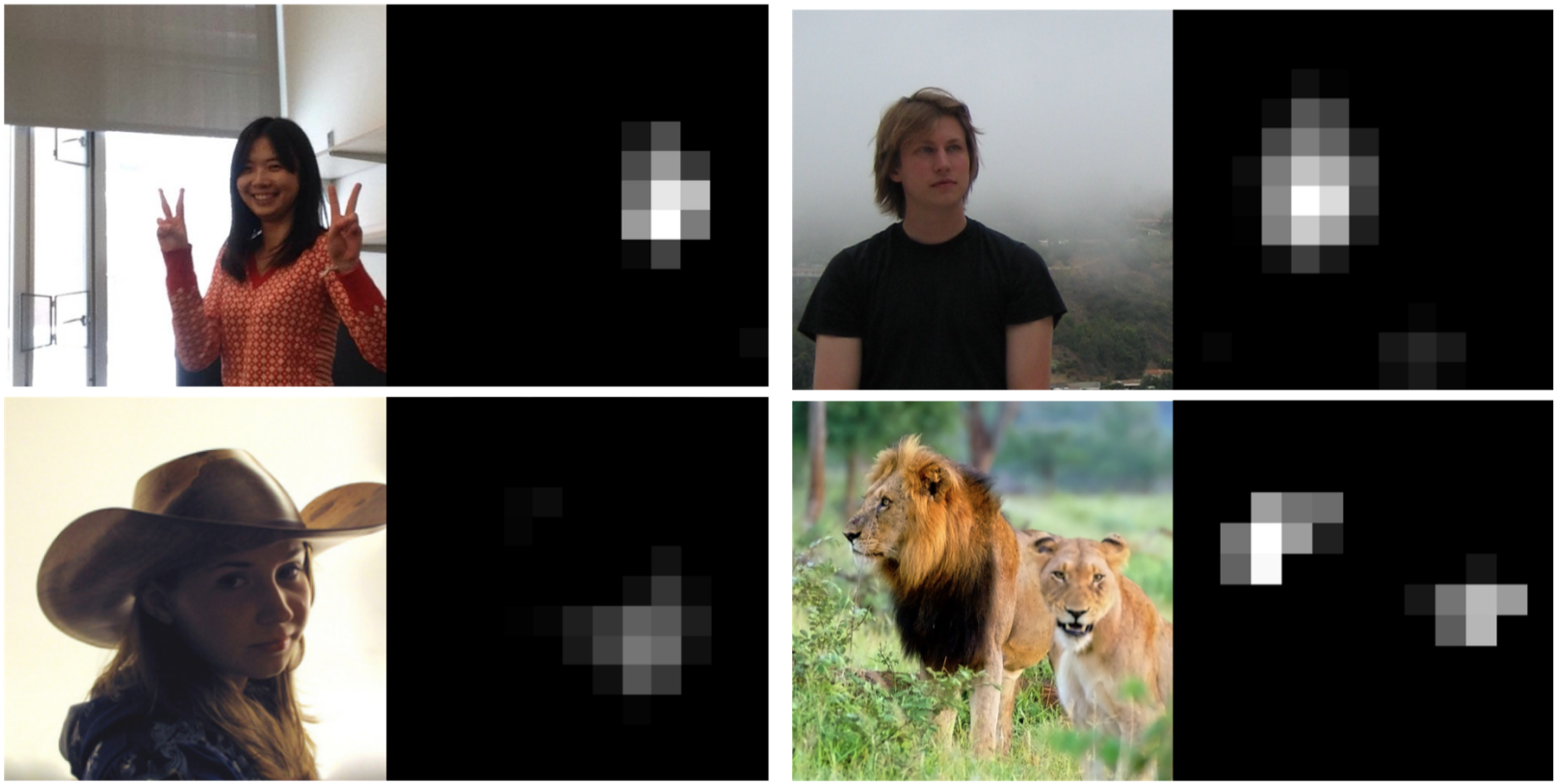
Emergent higher level abstractions

- Look at output of filter in 5th layer!



Emergent higher level abstractions

- Look at output of filter in 5th layer!



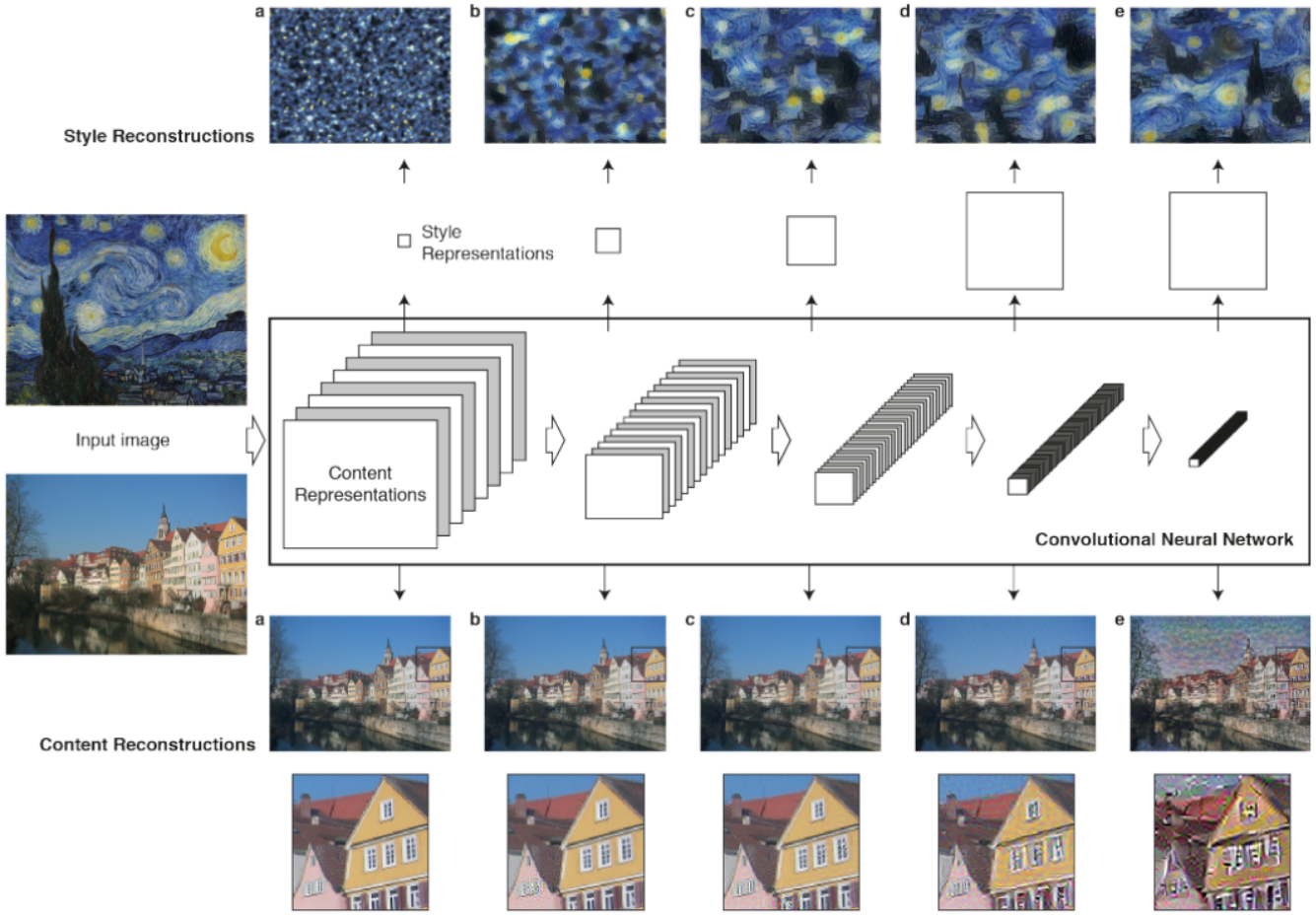
Yosinski et. al., ICML, google: deepvis

Google Deep Dream

http://deepdreamgenerator.com/

A Neural Algorithm of Artistic Style

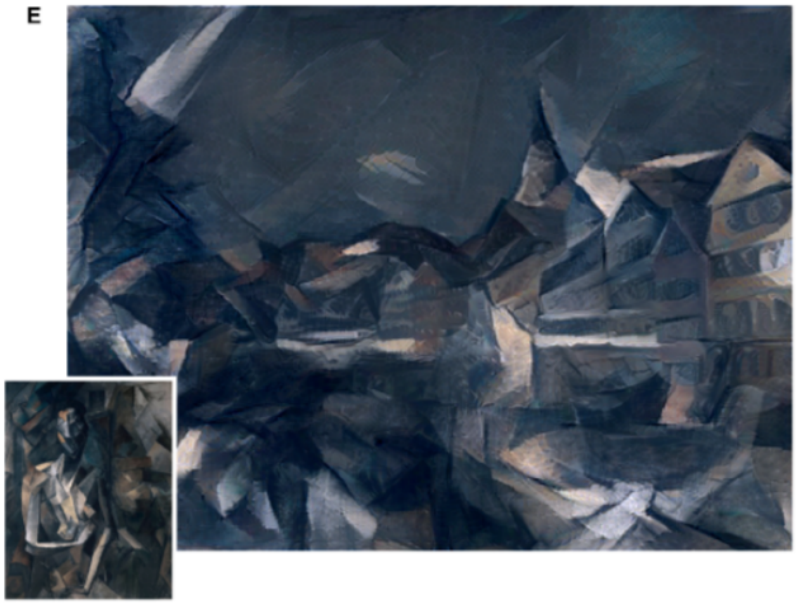
- Gatys, Ecker and Bethge
  `http://arxiv.org/pdf/1508.06576v2.pdf`
- Idea: Separate content and style



Borrowing the style from the masters!

Borrowing the style from the masters!

Borrowing the style from the masters!



Neural artistic style - The Movie

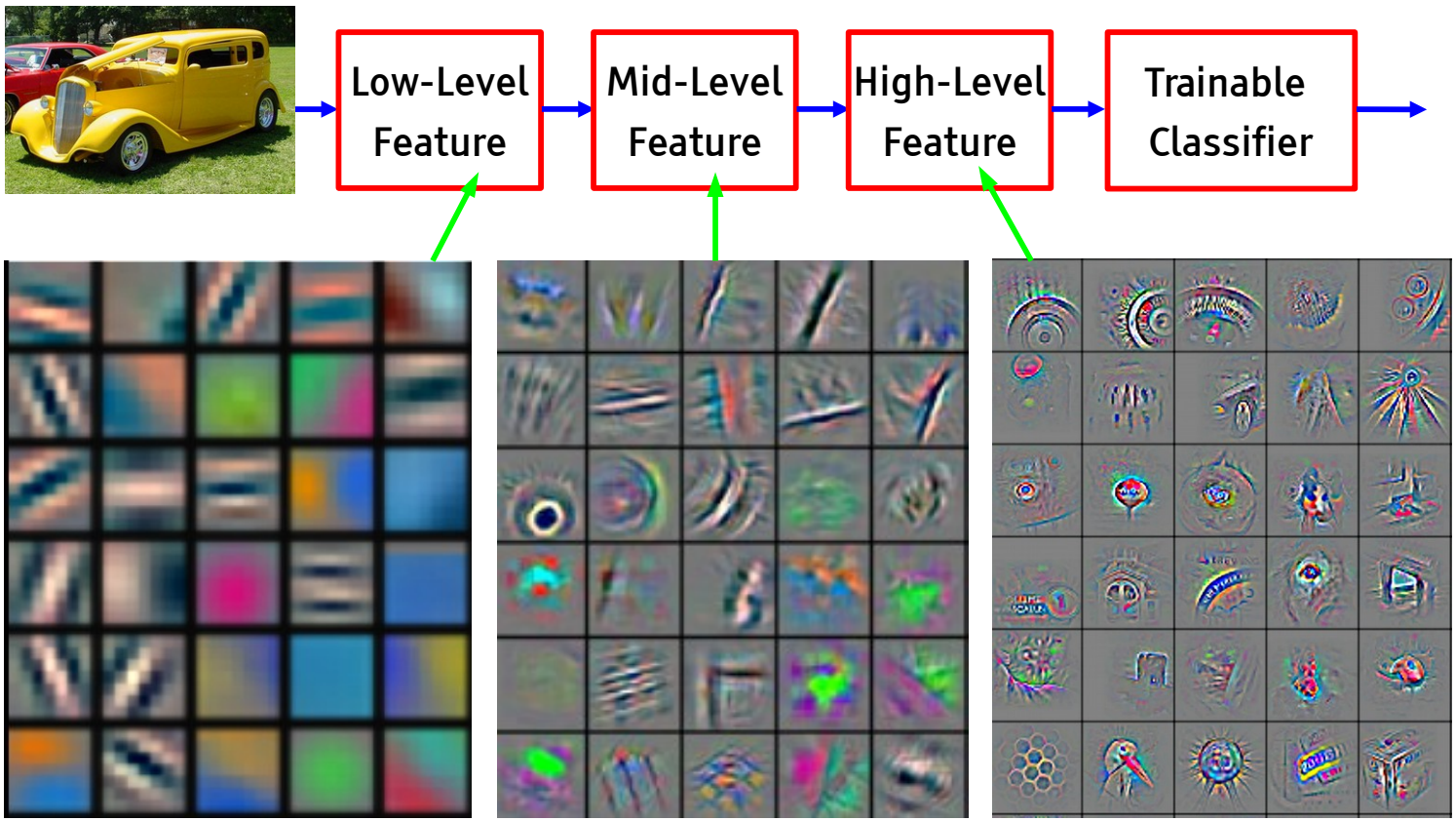
Part 2:

Convolutional NNs
The details

Deep Learning = Learning Hierarchical Representations

Y LeCun

- It's deep if it has more than one stage of non-linear feature transformation

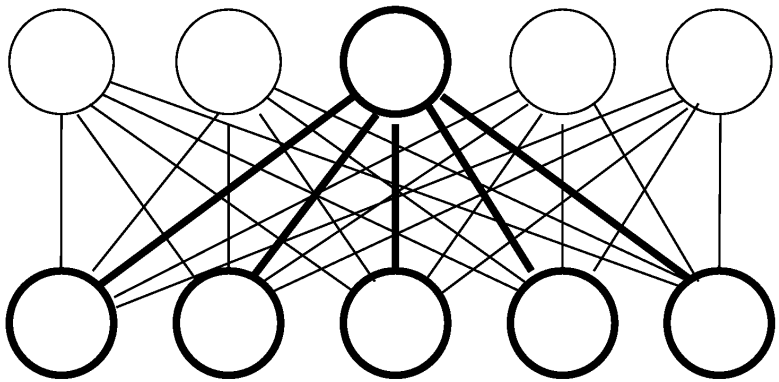


Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

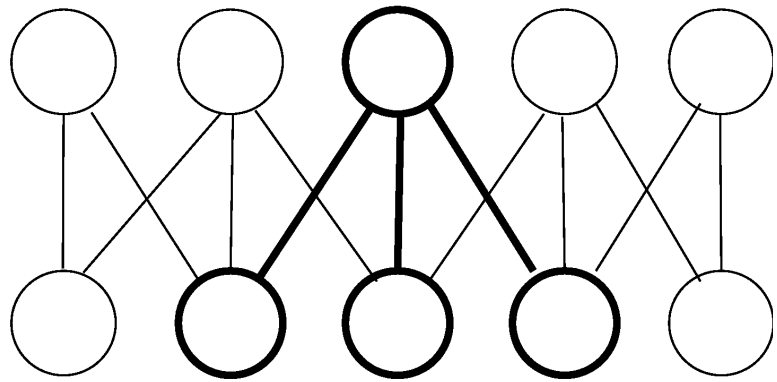
Convolutional Networks (LeCun et al., 1989)

- How would MNIST intro network scale to real images?
- Instead of hundreds of pixels, a million.
- Would weight matrices $\mathbf{W}$ be thousands $\times$ million?

Local connectivity



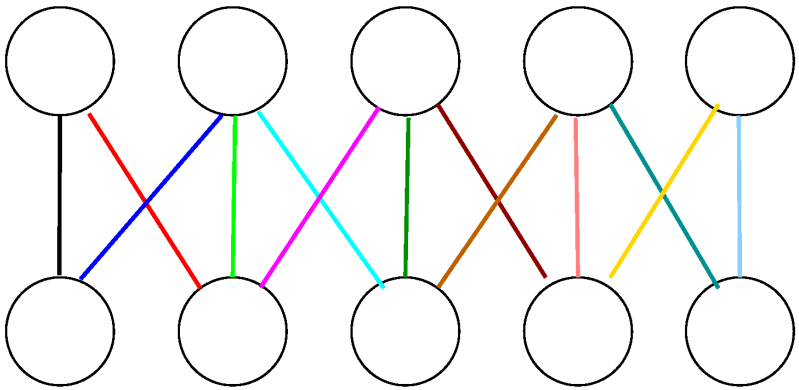
25 parameters



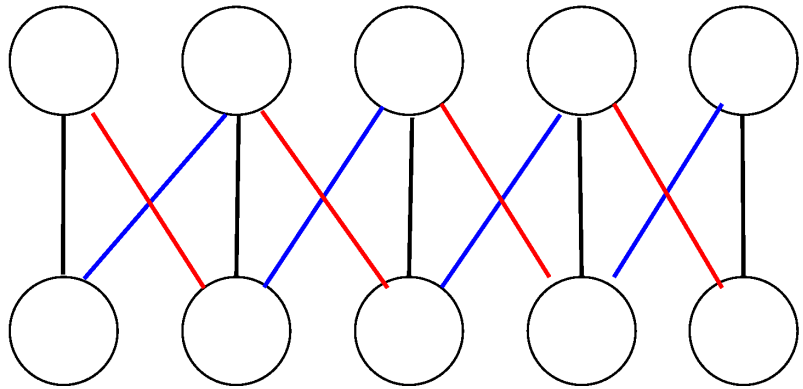
13 parameters

- Weight is zero unless $|x_1 - x_2| \leq k$ and $|y_1 - y_2| \leq k$

Parameter sharing



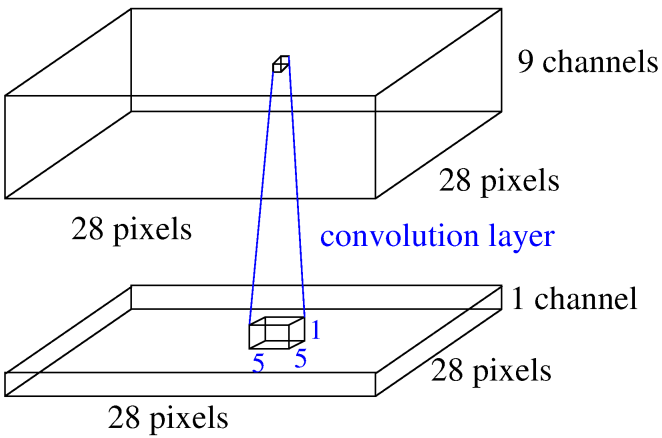
13 parameters

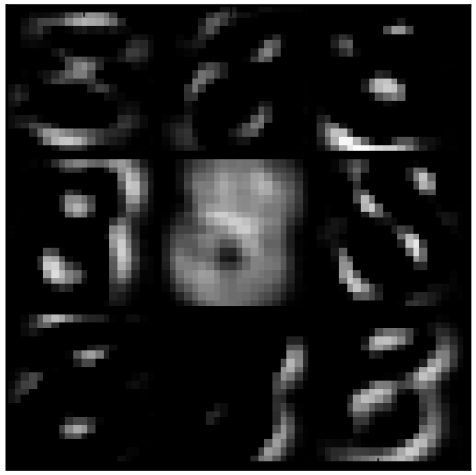


3 parameters

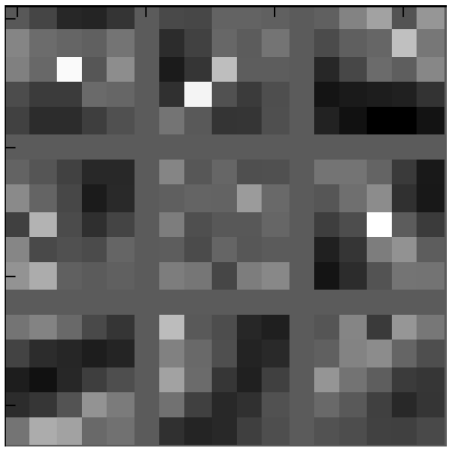
- Weights indexed by $\Delta x = x_1 - x_2$ and $\Delta y = y_1 - y_2$.
- $W_{\Delta x, \Delta y, c_1, c_2}$ replaces $W_{x_1, x_2, y_1, y_2, c_1, c_2}$.

Example (MNIST again)

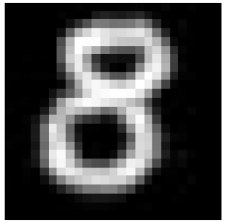




signals

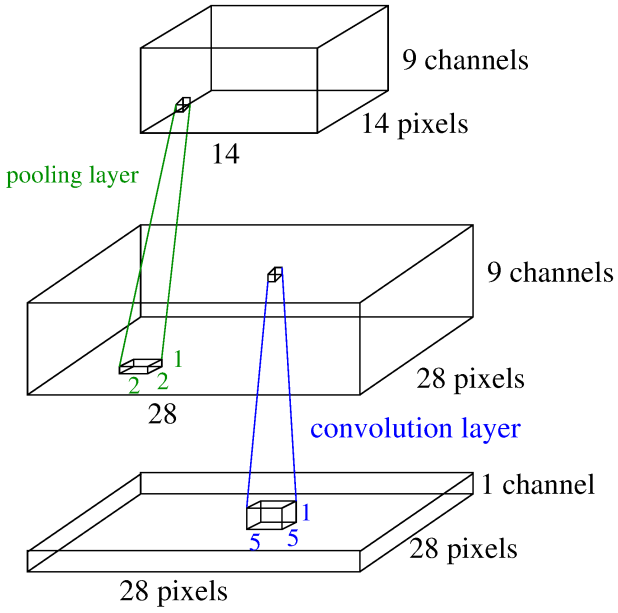


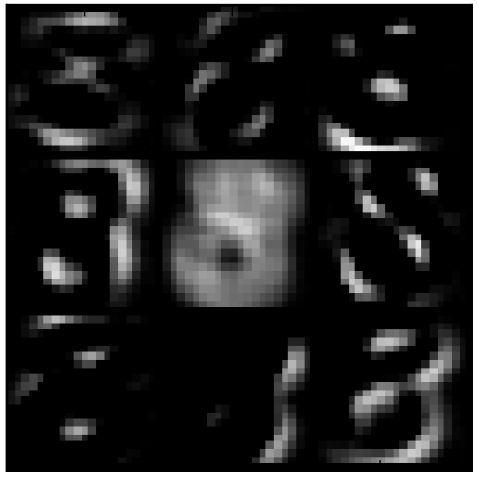
weights

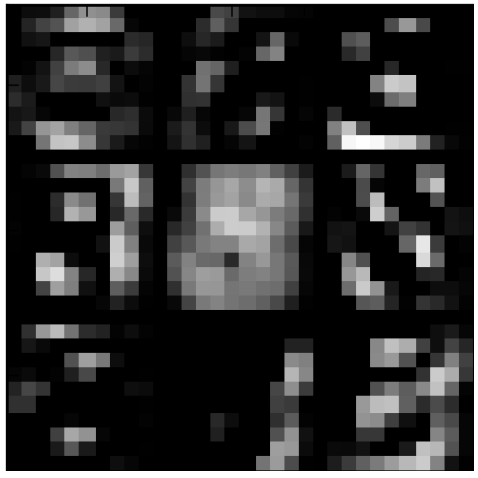


- MNIST data is 28×28 pixels and 1 channel.
- First hidden layer has 28×28 pixels and 9 channels.
- Use 5×5 filters ($\Delta x \leq 2$ and $\Delta y \leq 2$).

Pooling

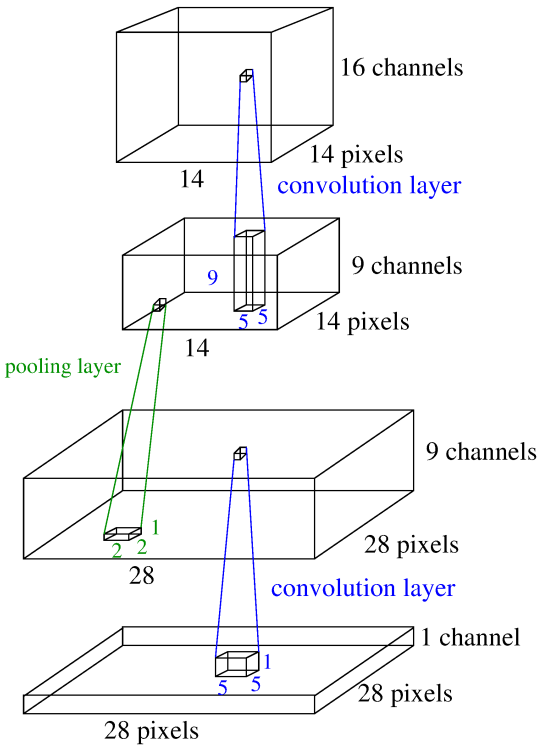


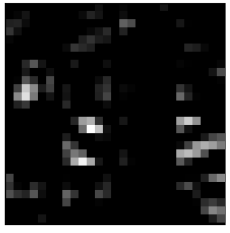


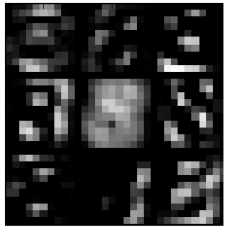


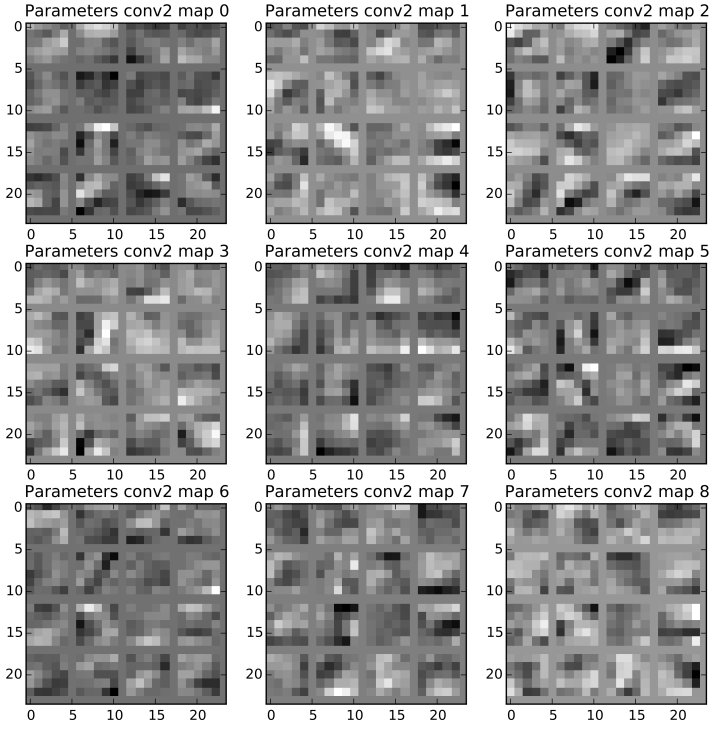
- Decrease resolution and increase channels going up.
- Often down-sampling by hard-coded pooling layers.
- We use $\max(\cdot)$ of 2×2 activations.
- Analysis of pooling functions (Boureau et al., 2010)

Stack more layers



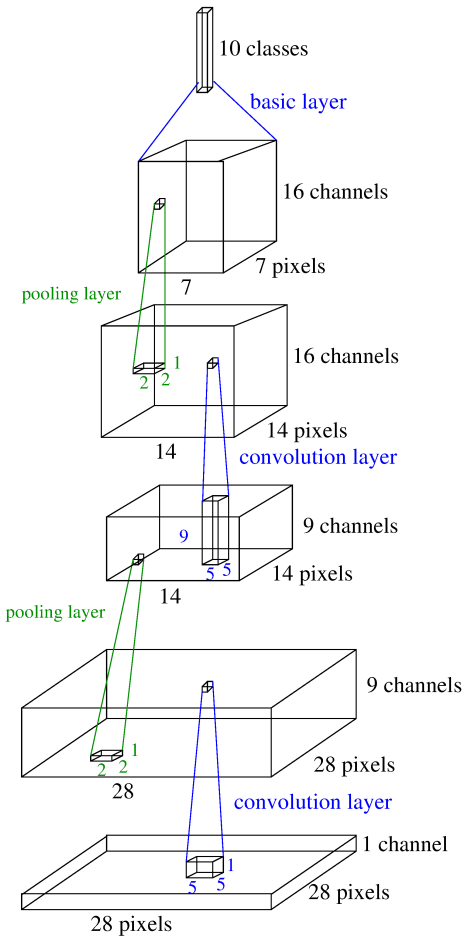


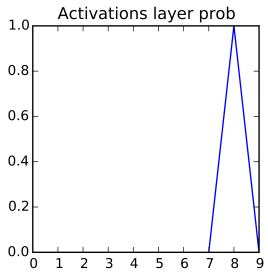


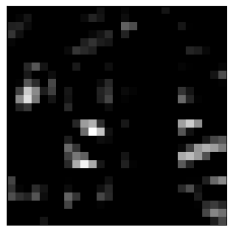


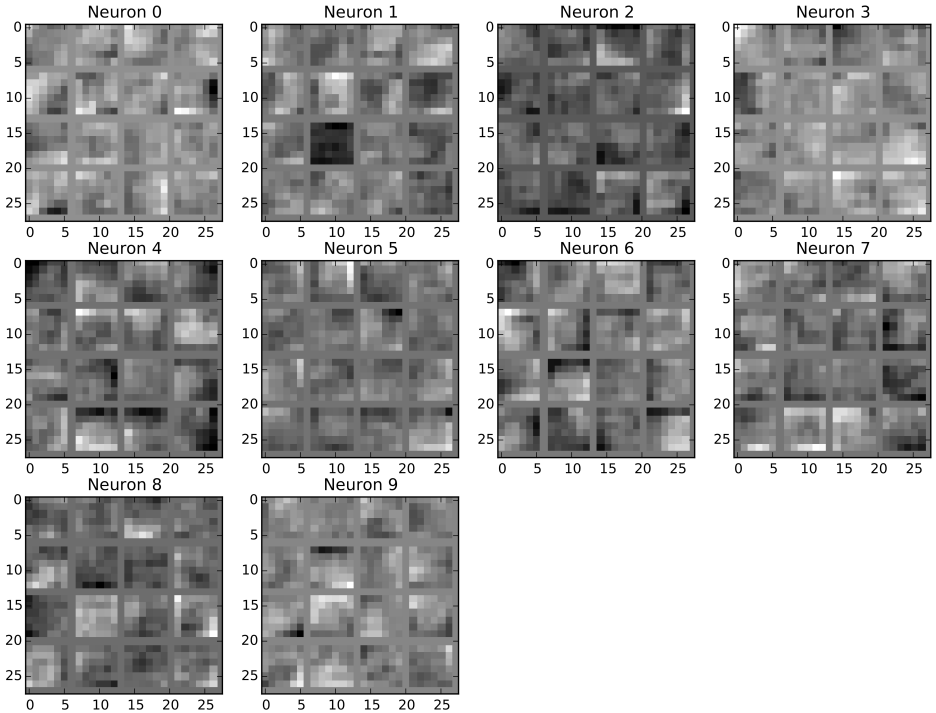
- Note how each unit looks at all channels of the previous layer.

Whole network









- At the top, the resolution drops to
- 1×1 pixel.
- Train by back-propagation as before.

Convolutional vs. non-convolutional

- Number of weights (ignoring biases):

  $5 \cdot 5 \cdot 9 + 5 \cdot 5 \cdot 9 \cdot 16 + 7 \cdot 7 \cdot 16 \cdot 10 = 225 + 3600 + 7840 = 11665$
- Sizes of signals $\mathbf{h}$:

  $28 \cdot 28, 28 \cdot 28 \cdot 9, 14 \cdot 14 \cdot 16 = 784, 7056, 3136$

Convolutional vs. non-convolutional

- Number of weights (ignoring biases):

  $5 \cdot 5 \cdot 9 + 5 \cdot 5 \cdot 9 \cdot 16 + 7 \cdot 7 \cdot 16 \cdot 10 = 225 + 3600 + 7840 = 11665$
- Sizes of signals $\mathbf{h}$:

  $28 \cdot 28, 28 \cdot 28 \cdot 9, 14 \cdot 14 \cdot 16 = 784, 7056, 3136$

- Compare to the introductory MNIST FFNN example
- Weights:

  $28 \cdot 28 \cdot 225 + 225 \cdot 144 + 144 \cdot 10 =$
  $176400 + 32400 + 1440 = 210240$
- Signals:

  $784, 225, 144$

- Convolutional network has more signals but less params.

Latest trends in CNNs

- New CNN architectures introduced in 2016-2017.
- Image segmentation - important application for autonomous systems and medical imaging
- New activation functions
- See extra slides

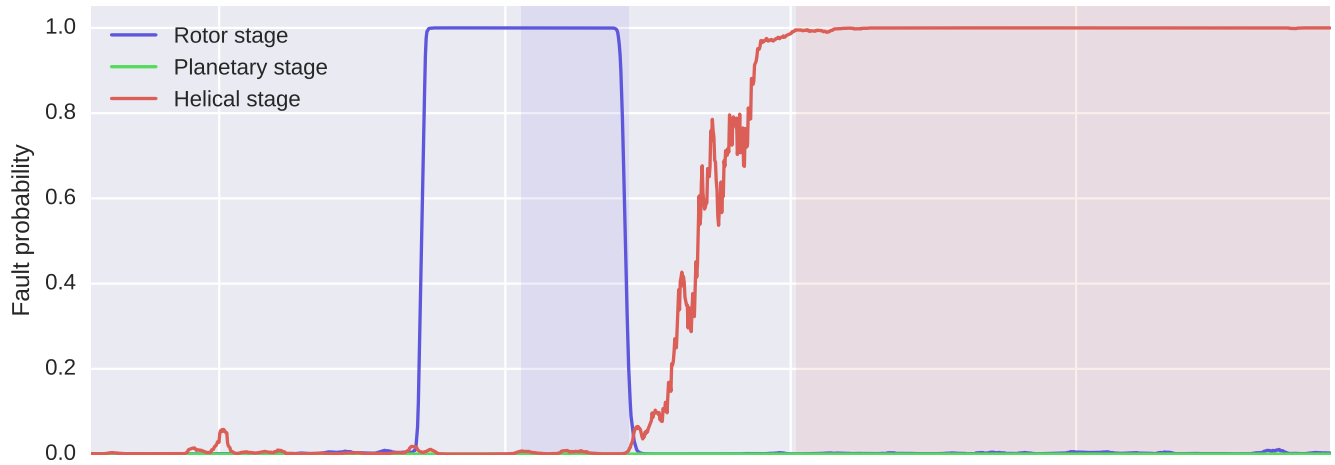
Part 3:
Condition monitoring

Industrial PhD with Siemens Wind Power

- Data is there
  - 10k turbines monitored for 5+ years with
  - detailed vibration and other sensor data
  - 100s of faults of different types
- Organisation already taking a data-driven approach.

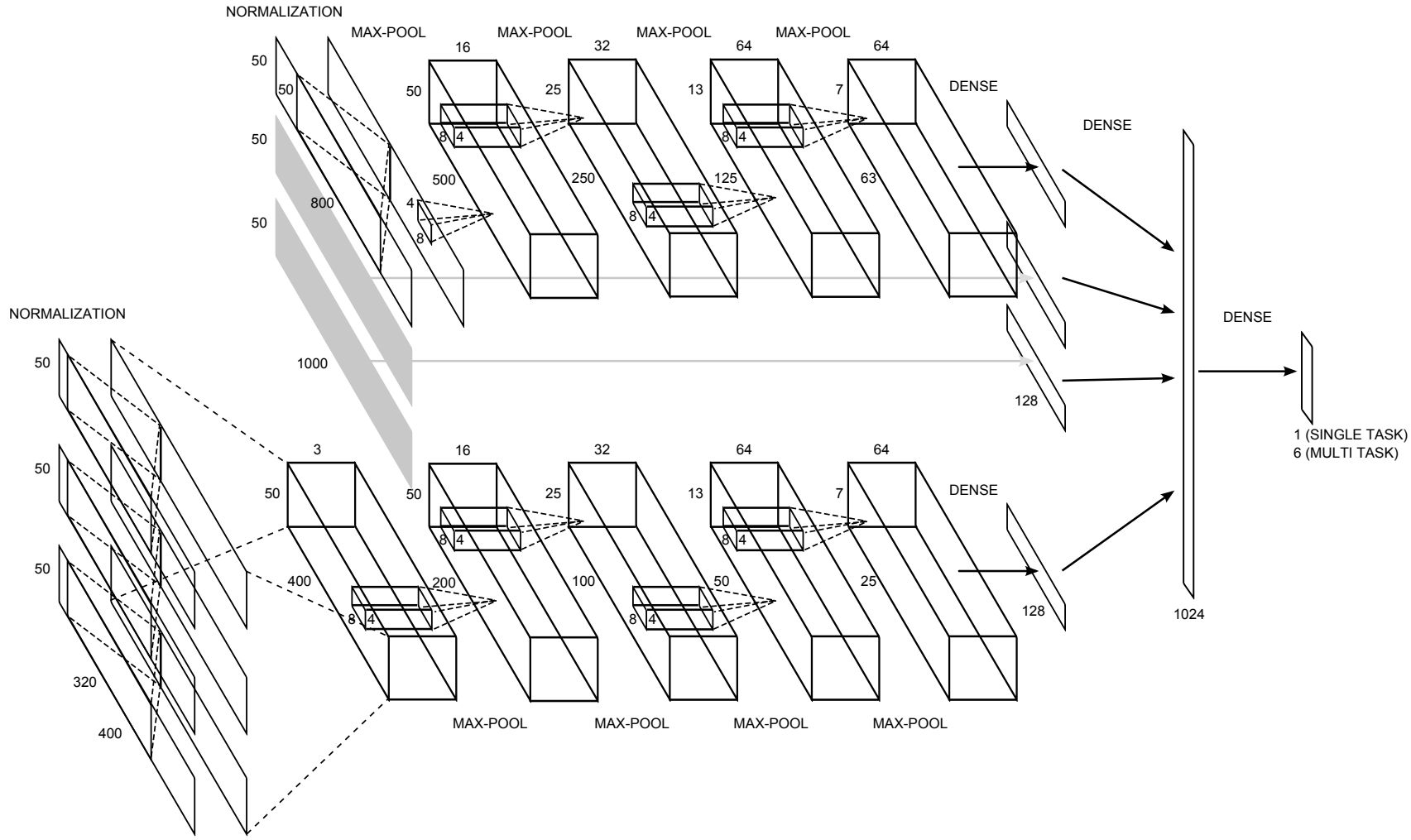
Industrial PhD with Siemens Wind Power

- Data is there
  - 10k turbines monitored for 5+ years with
  - detailed vibration and other sensor data
  - 100s of faults of different types
- Organisation already taking a data-driven approach.

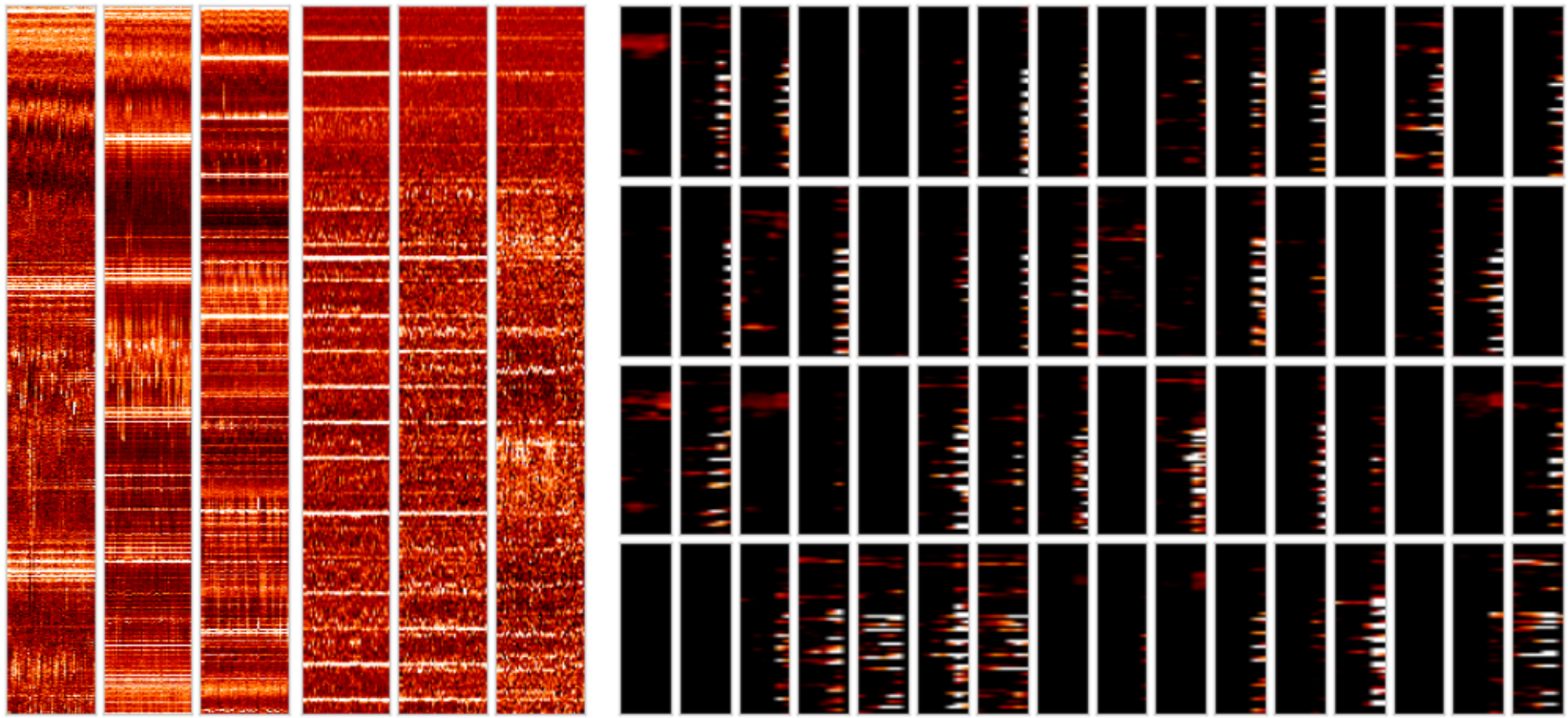


- Work by Martin Bach-Andersen, in Wind Energy, 2018

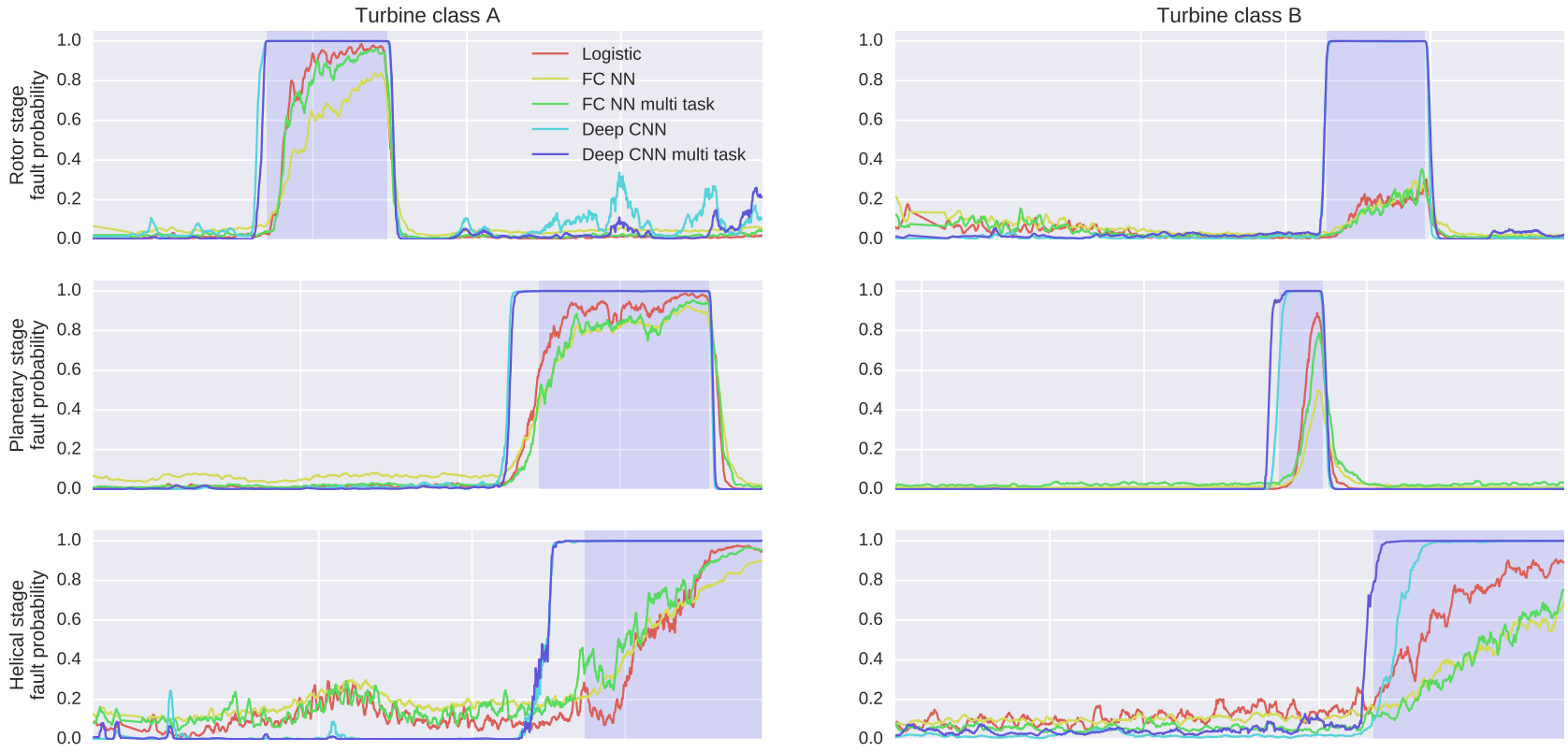
Convolutional neural network for vibrational data



Visualising how the network processes data



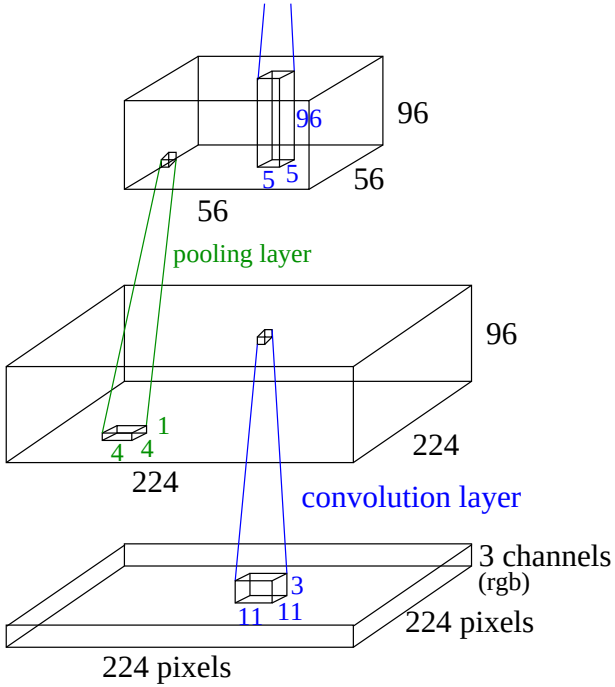
We really need this complicated architecture



Part 4:

Res-, Dense- and WaveNets

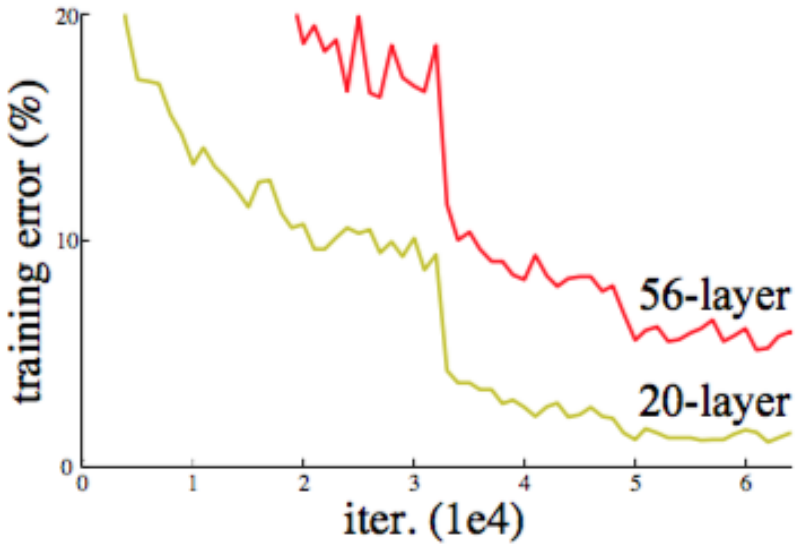
Scaling up

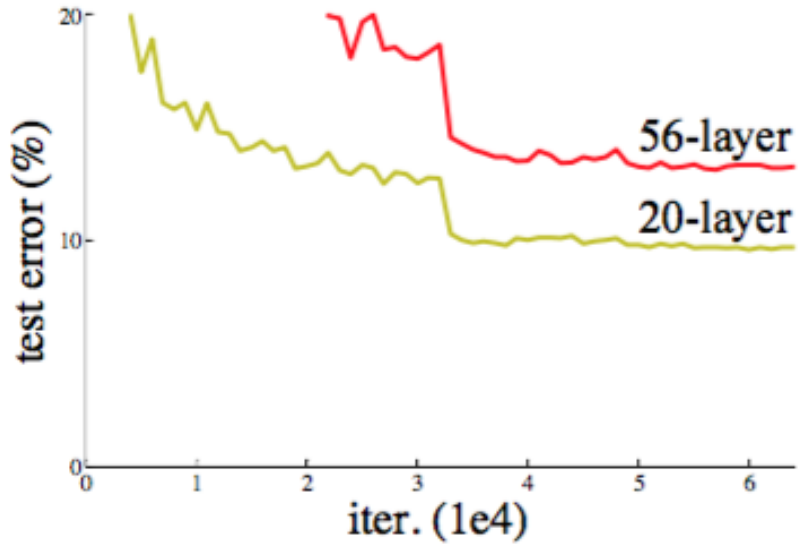


- First layers of (Krizhevsky, 2012).
- Trend in 2015 towards small filters (3×3) and poolings (2×2),
- but lots of layers (10–40).
- 2015+ trends:
  - Even smaller filters
  - sometimes 1×1,
  - more layers and
  - average pooling in transition from convolutional to fully connected layers.

Deep residual nets

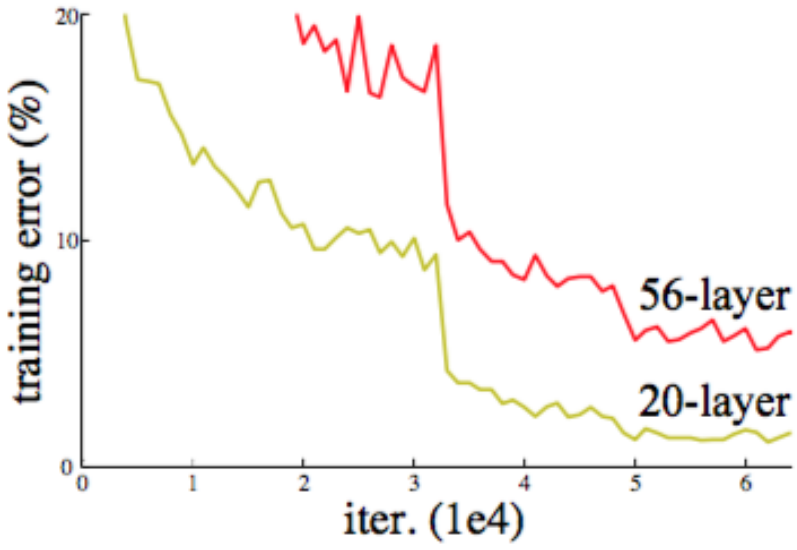
- Observation - it is difficult to fit really deep nets:

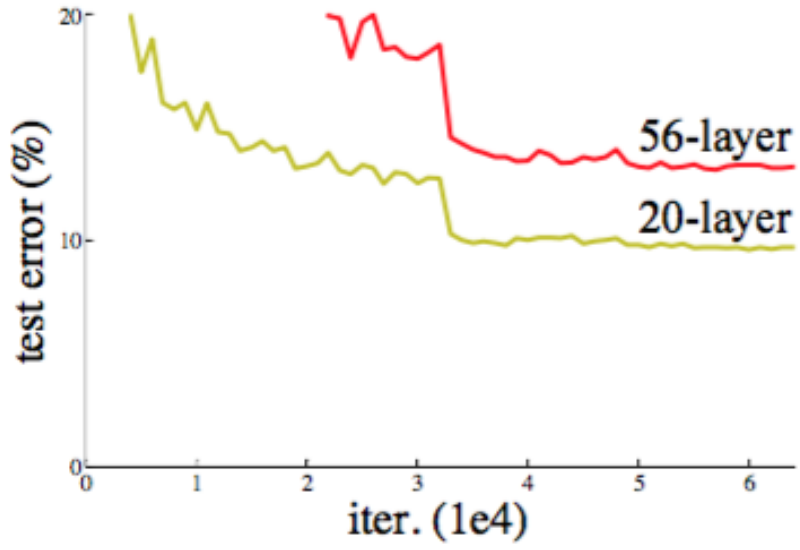




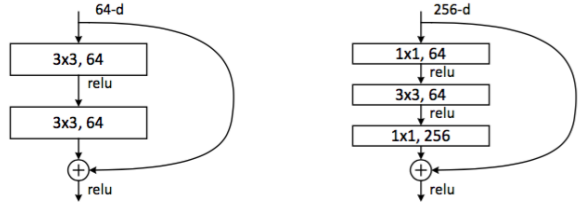
Deep residual nets

- Observation - it is difficult to fit really deep nets:



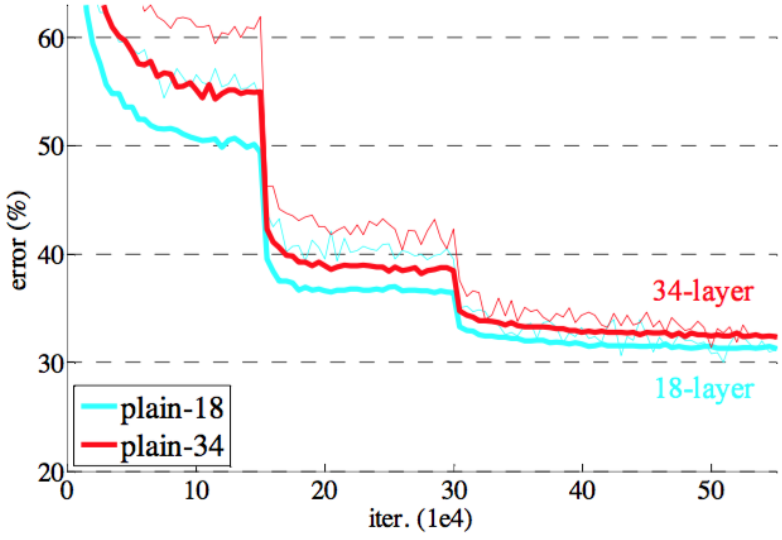


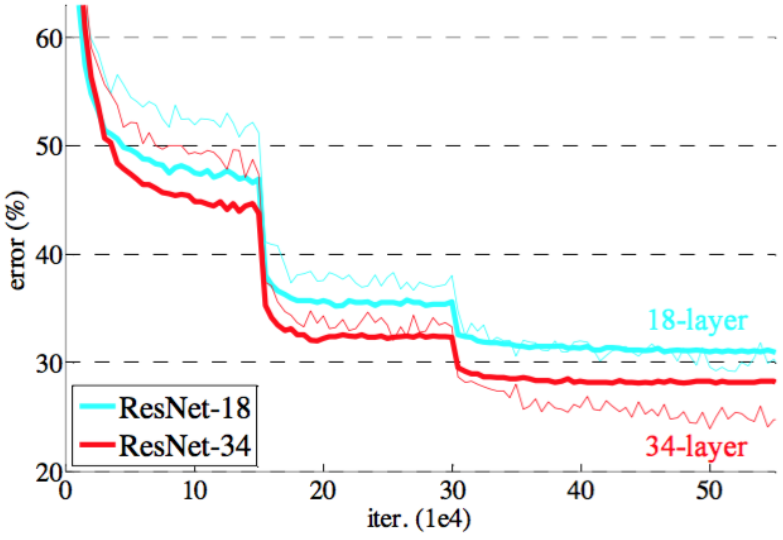
- Solution - bypass layers: $\mathbf{h}_l = \mathcal{F}(\mathbf{h}_{l-1}) + \mathbf{h}_{l-1}$



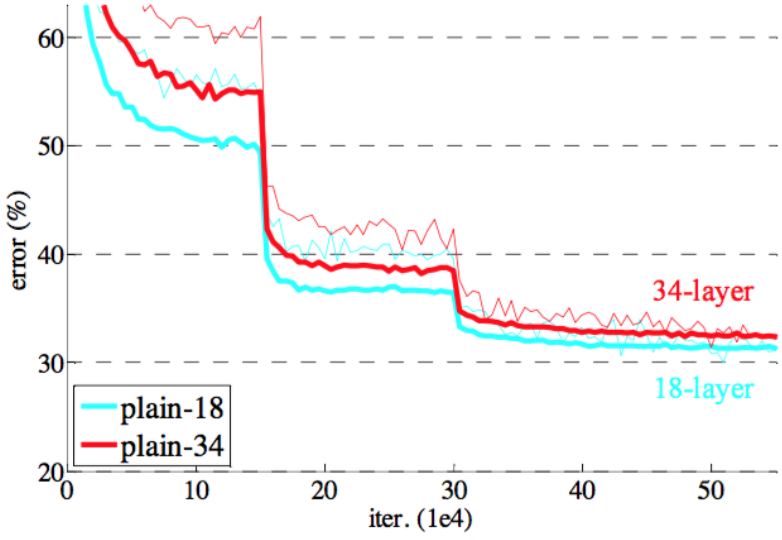
- Initially $\mathcal{F}(\mathbf{h}_{l-1}) \approx 0$ and net $\approx$ linear

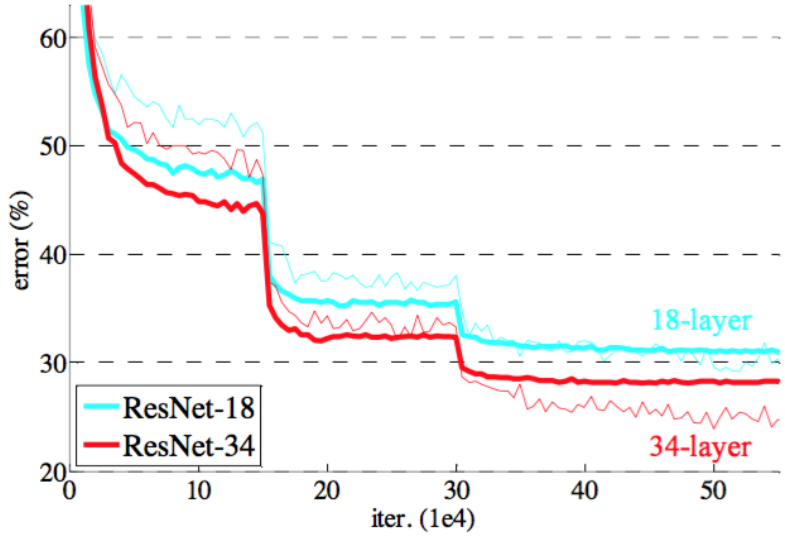
Deep residual nets results - ImageNet results





Deep residual nets results - ImageNet results

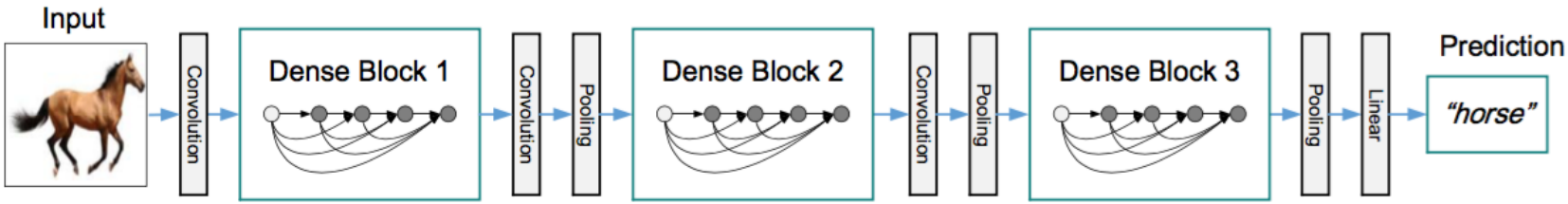




method	top-1 err.	top-5 err.
VGG [41] (ILSVRC'14)	-	8.43†
GoogLeNet [44] (ILSVRC'14)	-	7.89
VGG [41] (v5)	24.4	7.1
PReLU-net [13]	21.59	5.71
BN-inception [16]	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60
ResNet-50	20.74	5.25
ResNet-101	19.87	4.60
ResNet-152	**19.38**	**4.49**

- 10 ensemble models: ResNets 3.57% top 5 error.
- GoogleNet (2014) 6.66%

DenseNets



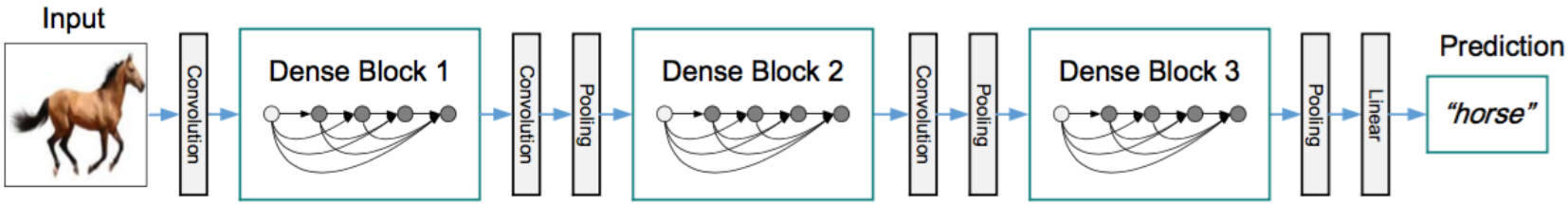
- Reuse features extracted in previous layers

$$\mathbf{h}_l = \mathcal{F}([\mathbf{h}_0, \ldots, \mathbf{h}_{l-1}])$$

- Compare this with ResNets

$$\mathbf{h}_l = \mathcal{F}(\mathbf{h}_{l-1}) + \mathbf{h}_{l-1}$$

DenseNets



- Reuse features extracted in previous layers

$$\mathbf{h}_l = \mathcal{F}([\mathbf{h}_0, \ldots, \mathbf{h}_{l-1}])$$

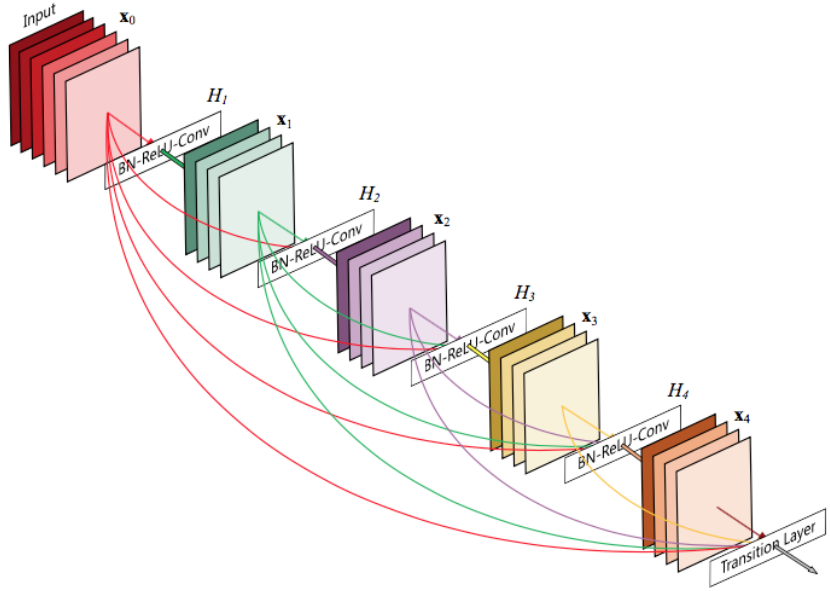
- Compare this with ResNets

$$\mathbf{h}_l = \mathcal{F}(\mathbf{h}_{l-1}) + \mathbf{h}_{l-1}$$

- Avoids parameter explosion by: (details next slide)
  - Having few filters in each layer and
  - use of transition layers
- State-of-the-art on CIFAR and SVHN
- (not tested on ImageNet)
- with fewer parameters.

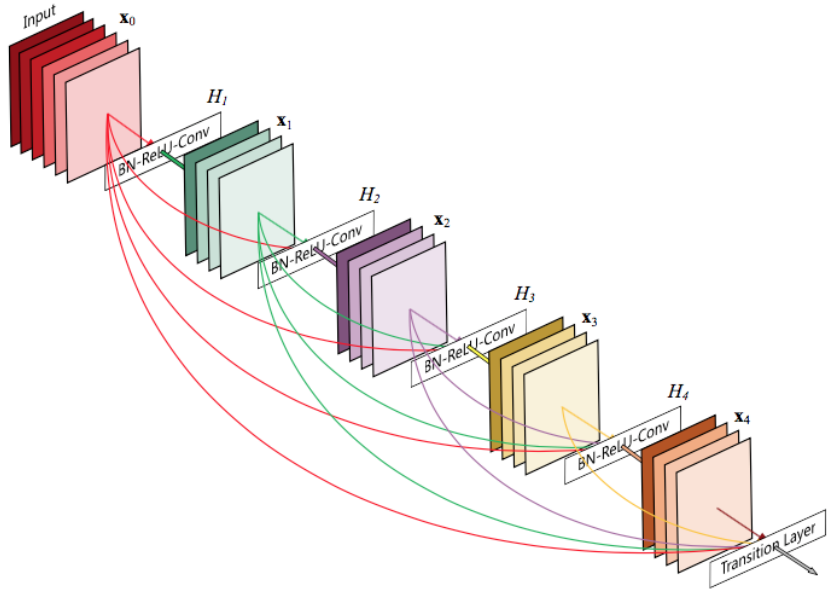
DenseNets details

- L layers in a dense block - $L = 5$ in figure
- Growth factor (feature maps in a layer) k - $k = 4$ in figure
- Feature maps are connected to all subsequent layers in subsequent layers: $l \times k$ channels in layer l.
- Total $\frac{L(L+1)}{2}k$ vs LK in standard.



DenseNets details

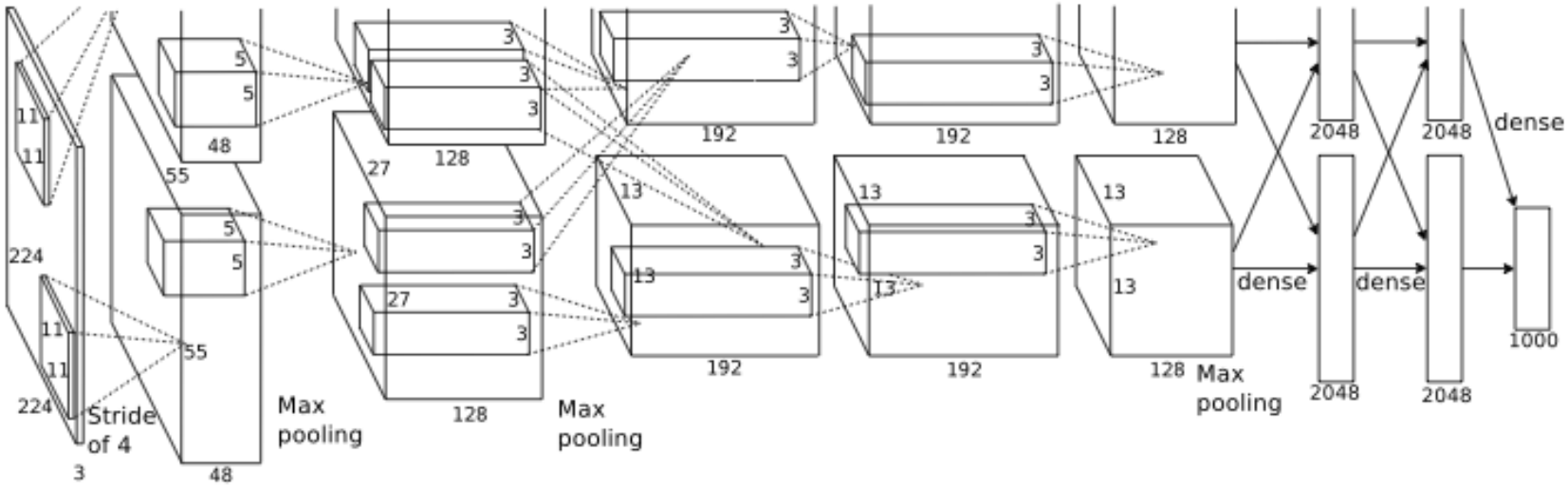
- L layers in a dense block - $L = 5$ in figure
- Growth factor (feature maps in a layer) k - $k = 4$ in figure
- Feature maps are connected to all subsequent layers in subsequent layers: $l \times k$ channels in layer l.
- Total $\frac{L(L+1)}{2}k$ vs LK in standard.



- Keep number of parameters by small k and
- use transition layers (with 1×1 convolutions and pooling)
- between dense blocks.

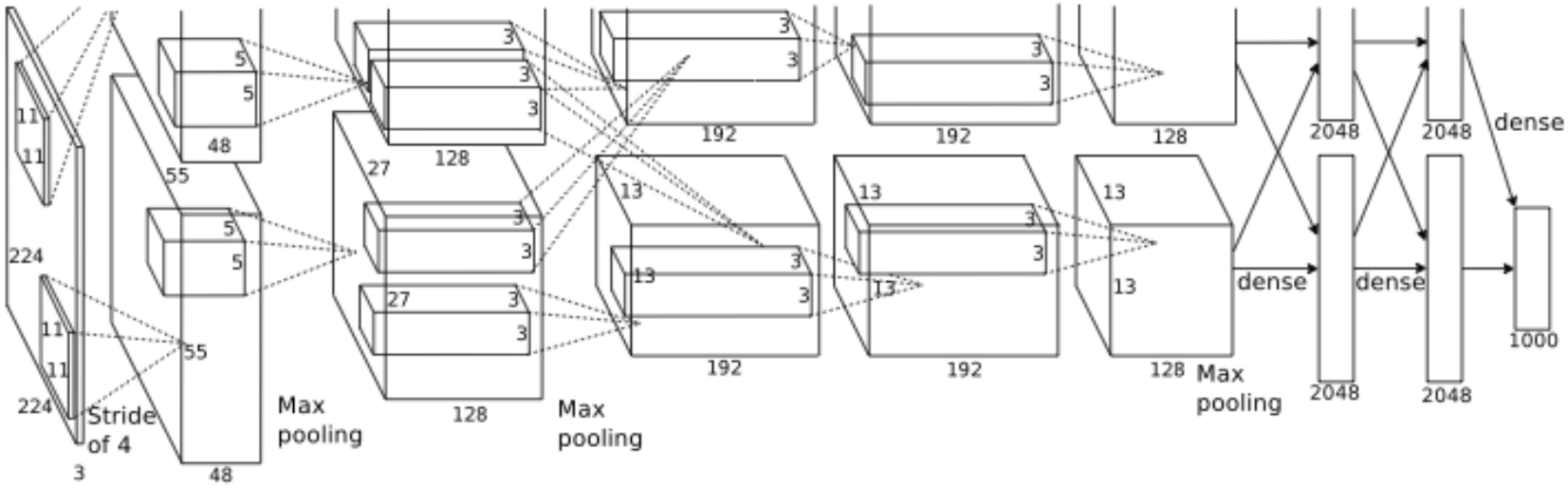
We need bigger brains

- AlexNet (2012): 16.4% error, 8 layers, 1.4 Gflop

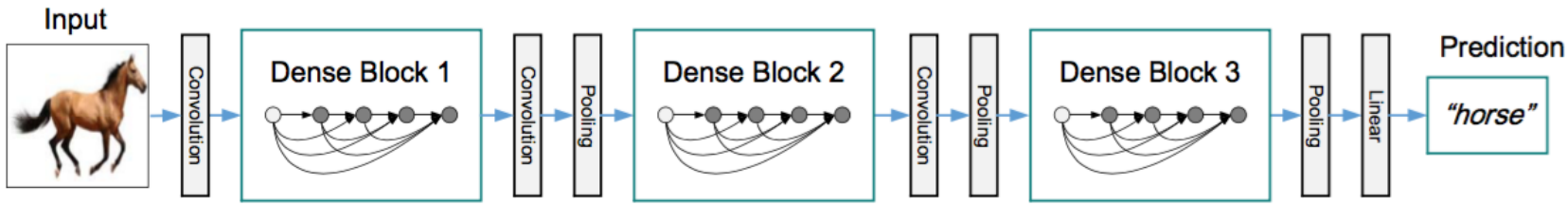


We need bigger brains

- AlexNet (2012): 16.4% error, 8 layers, 1.4 Gflop

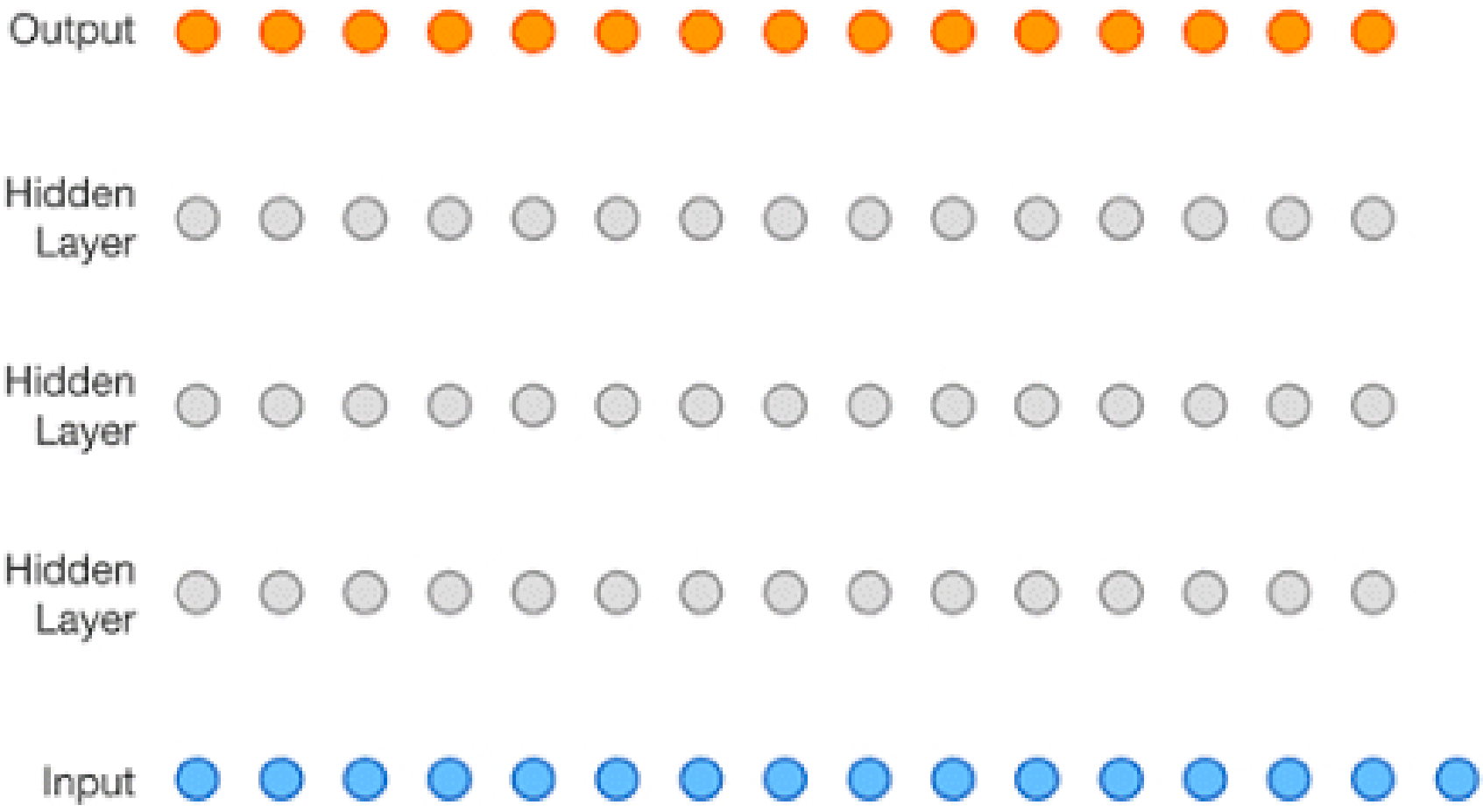


- ResNet (2016): 3.5% error, 152 layers, 22.6 Gflop.



- (This is a so-called DenseNet and not a ResNet.)
- Source: Source Jen-Hsun Huang, CEO NVIDIA, GTC Europe, 2016

WaveNet



DeepMind blogpost and

`https://arxiv.org/pdf/1609.03499.pdf`

Part 5
New activation functions

Activation functions

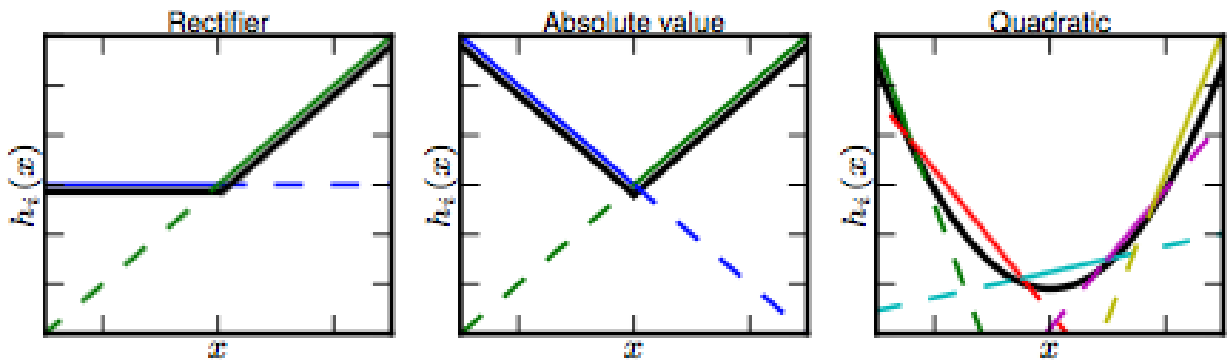
- Gated linear units (GLU) works well with conv nets

$$\mathrm{GLU}(x) = (Wx + b) \otimes \sigma(Vx + c)$$

- MaxOut for unit i. Each unit has K inputs

$$\mathrm{MaxOut}_i(x) = \max_{k \in 1,\ldots,K} z_{ik}$$

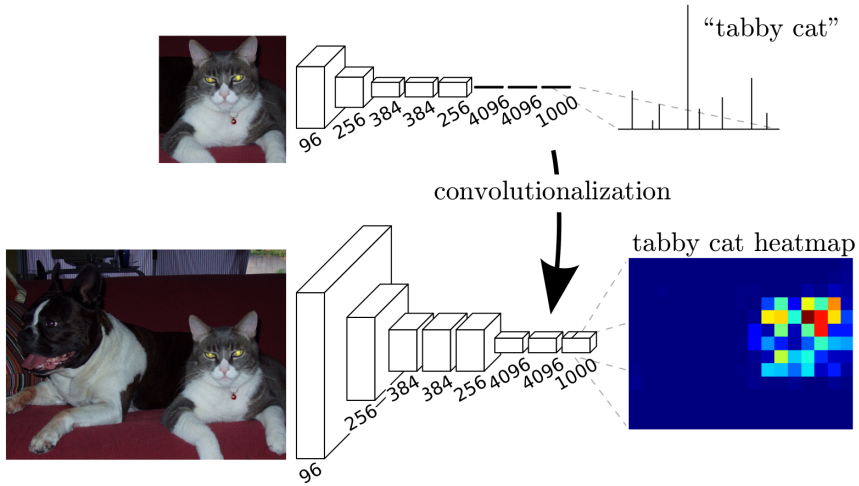
- Works well with dropout

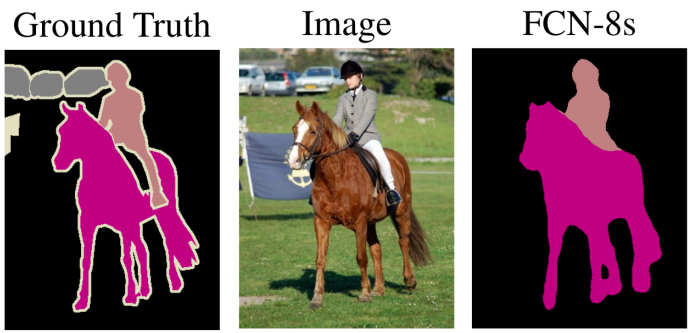


- $\mathrm{LeakyRelu}(z) = \max(az, z)$ has
- slope $a \in [0, 1]$ for $z < 0$
- Other in ReLu family: SoftPLus, ELU, SELU, etc

Part 6:
Image Segmentation

Semantic segmentation (Long et al., 2015)

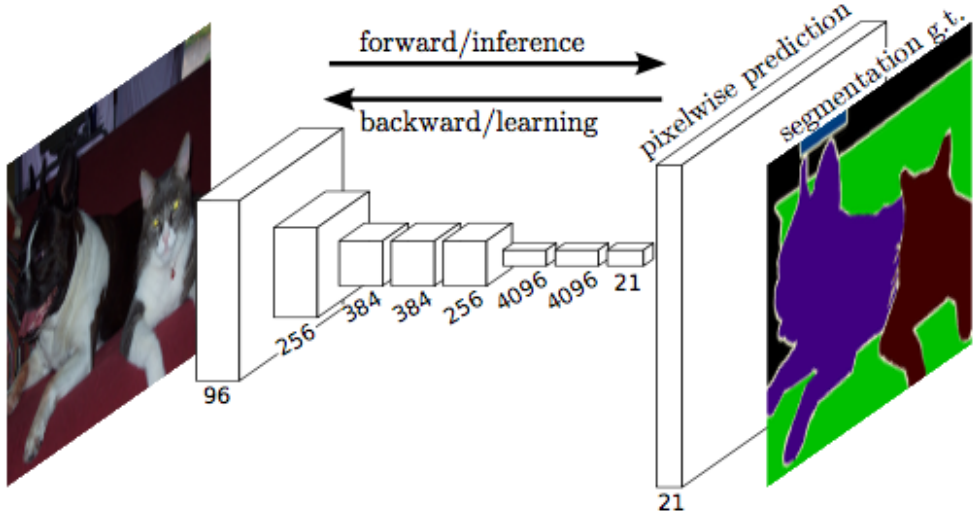




- Sliding a convolutional network to classify each location.
- Lots of shared computation.

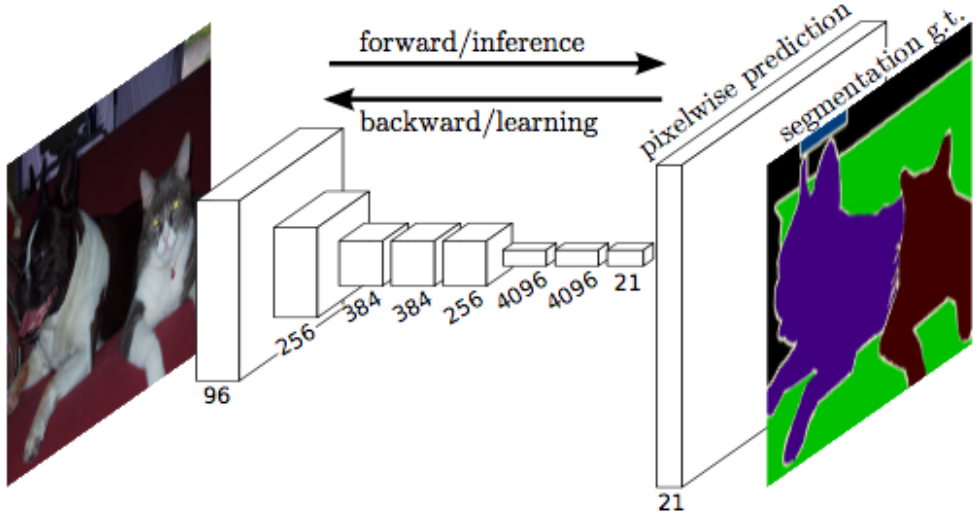
End-to-end image segmentation

- Classify each pixel

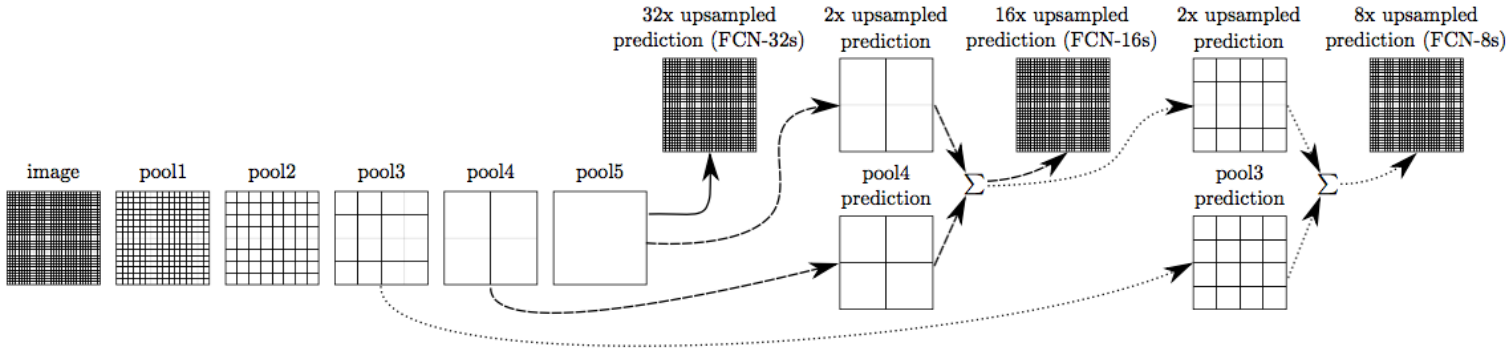


End-to-end image segmentation

- Classify each pixel

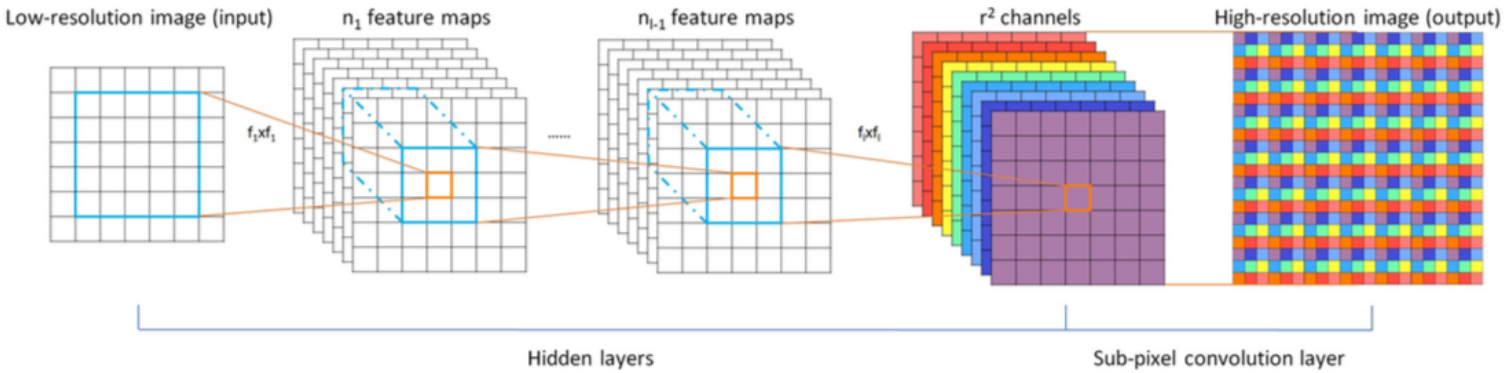


- Use convolutions, pooling and upsampling

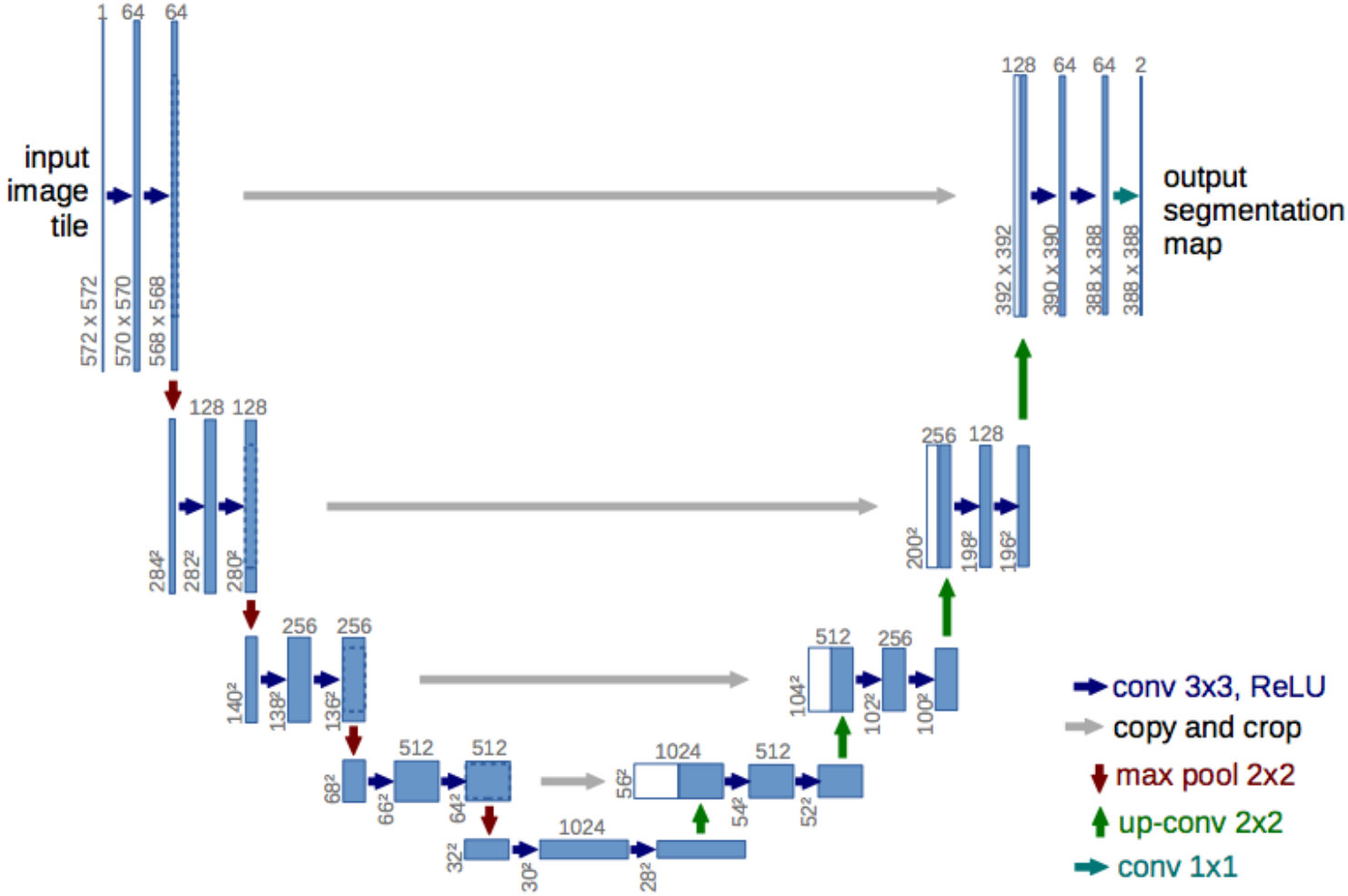


Sub-pixel deconvolutions

- Convert $H \times H \times C \rightarrow 2H \times 2H \times C/4$:



U-Net


input image tile
572 x 572
570 x 570
568 x 568
1
64
64
128
128
284²
282²
280²
256
256
140²
138²
136²
512
512
68²
66²
64²
1024
32²
30²
28²
1024
512
56²
54²
512
256
104²
102²
100²
256
128
200²
198²
196²
128
64
64
2
392 x 392
390 x 390
388 x 388
388 x 388
output segmentation map
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1

The Cityscapes dataset

Cityscapes leaderboard

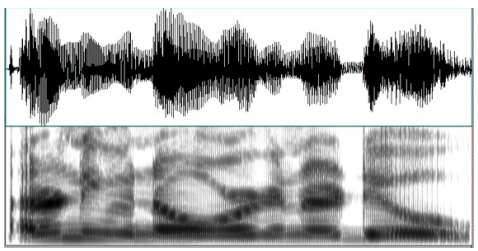
- Performance metric

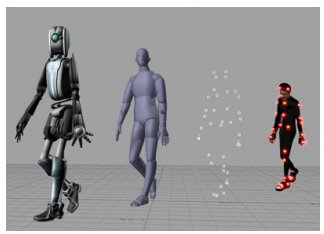
$$\text{Intersection over Union} = IoU = \frac{TP}{TP + FP + FN}$$

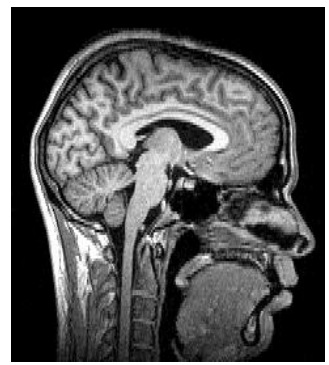
name	fine	coarse	16-bit	depth	video	sub	IoU class	iIoU class	IoU category	iIoU category	Runtime [s]	code
motovis	yes	yes	no	no	no	no	81.3	57.7	91.5	80.7	n/a	no
PSPNet	yes	yes	no	no	no	no	81.2	59.6	91.2	79.2	n/a	yes
NetWarp	yes	yes	no	no	yes	no	80.5	59.5	91.0	79.8	n/a	no
ResNet-38	yes	yes	no	no	no	no	80.6	57.8	91.0	79.1	n/a	yes
tek-Ifly	yes	no	no	no	no	no	81.1	60.1	90.9	79.6	n/a	no
ResNet-38	yes	no	no	no	no	no	78.4	59.1	90.9	81.1	n/a	yes
TuSimple_Coarse	yes	yes	no	no	no	no	80.1	56.9	90.7	77.8	n/a	yes
SAC-multiple	yes	no	no	no	no	no	78.1	55.2	90.6	78.3	n/a	no
SegModel	yes	yes	no	no	no	no	79.2	56.4	90.4	77.0	n/a	no
TuSimple	yes	no	no	no	no	no	77.6	53.6	90.1	75.2	n/a	no
Global-Local-Refinement	yes	no	no	no	no	no	77.3	53.4	90.0	76.8	n/a	no

Other applications

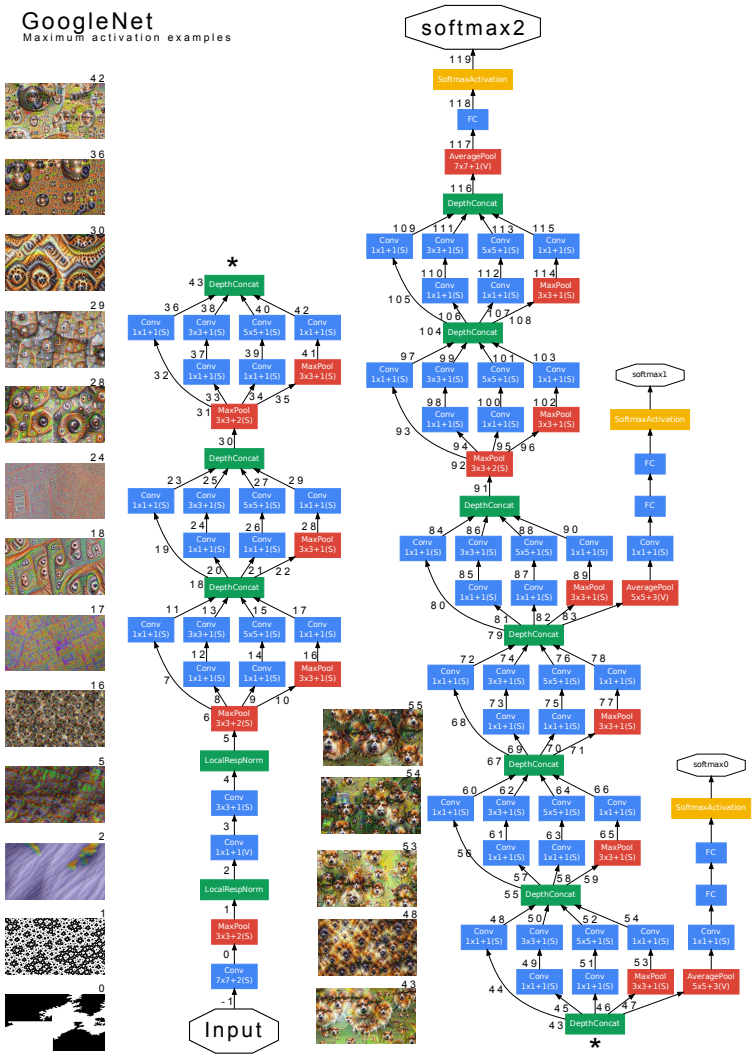
Tensor	Single channel	Multi-channel
1-D	Raw audio (mono)	Motion capture
2-D	Audio + Fourier transform	Game of Go
3-D	Brain imaging	Colour video

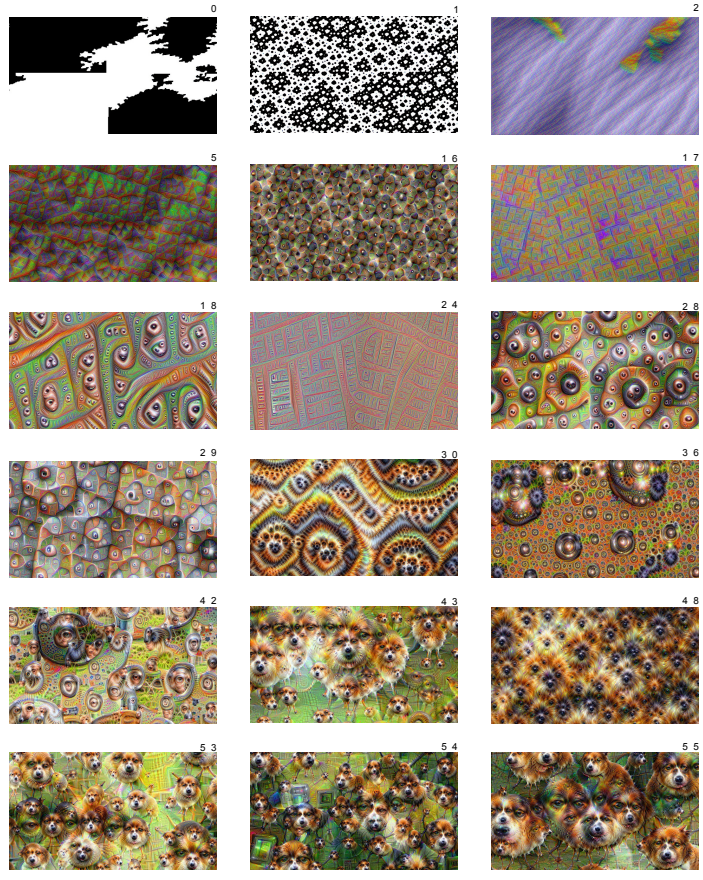






Inceptionism (Mordvintsev, 2015) Video by Miquel Perello Nieto





Links: Youtube video, image gallery and Google blog

References

- LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4), 541-551.
- Boureau, Y-Lan, Jean Ponce, and Yann LeCun. "A theoretical analysis of feature pooling in visual recognition." Proceedings of the 27th International Conference on Machine Learning (ICML-10). 2010.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).
- Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." arXiv preprint arXiv:1411.4038 (2015).
- Mathieu, Michael, Mikael Henaff, and Yann LeCun. "Fast training of convolutional networks through FFTs." arXiv preprint arXiv:1312.5851 (2013).
- Kivinen, Jyri J., and Christopher KI Williams (2011). "Transformation equivariant Boltzmann machines." Artificial Neural Networks and Machine Learning (ICANN).
- Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." Computer Vision–ECCV 2014. Springer International Publishing, 2014. 818-833.
- Mordvintsev, A., Olah, C., & Tyka, M. (2015) "Inceptionism: Going Deeper into Neural Networks", Google research blogspot
- He et al, Deep Residual Learning for Image Recognition, `https://arxiv.org/pdf/1512.03385v1.pdf`
- Huang et al, Densely connected convolutional networks, `http://arxiv.org/pdf/1608.06993.pdf`
- Oord et al, WaveNet a generative model for raw audio, `https://arxiv.org/pdf/1609.03499.pdf`

References on new trends

- He et al, Deep Residual Learning for Image Recognition, https://arxiv.org/pdf/1512.03385v1.pdf
- Huang et al, Densely connected convolutional networks, http://arxiv.org/pdf/1608.06993.pdf
- Oord et al, WaveNet a generative model for raw audio, https://arxiv.org/pdf/1609.03499.pdf
- Dauphin et al, Language Modeling with Gated Convolutional Networks, https://arxiv.org/pdf/1612.08083.pdf
- Goodfellow et al, MaxOut networks, https://arxiv.org/pdf/1302.4389.pdf
- Long et al, Fully convolutional networks for semantic segmentation, https://arxiv.org/abs/1411.4038
- Shi et al, Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network https://arxiv.org/abs/1609.05158
- Ronneberger et al, U-Net: Convolutional Networks for Biomedical Image Segmentation, https://arxiv.org/abs/1505.04597

Thanks!
Ole Winther