# Convolutional Networks

Tapani Raiko

Aalto University

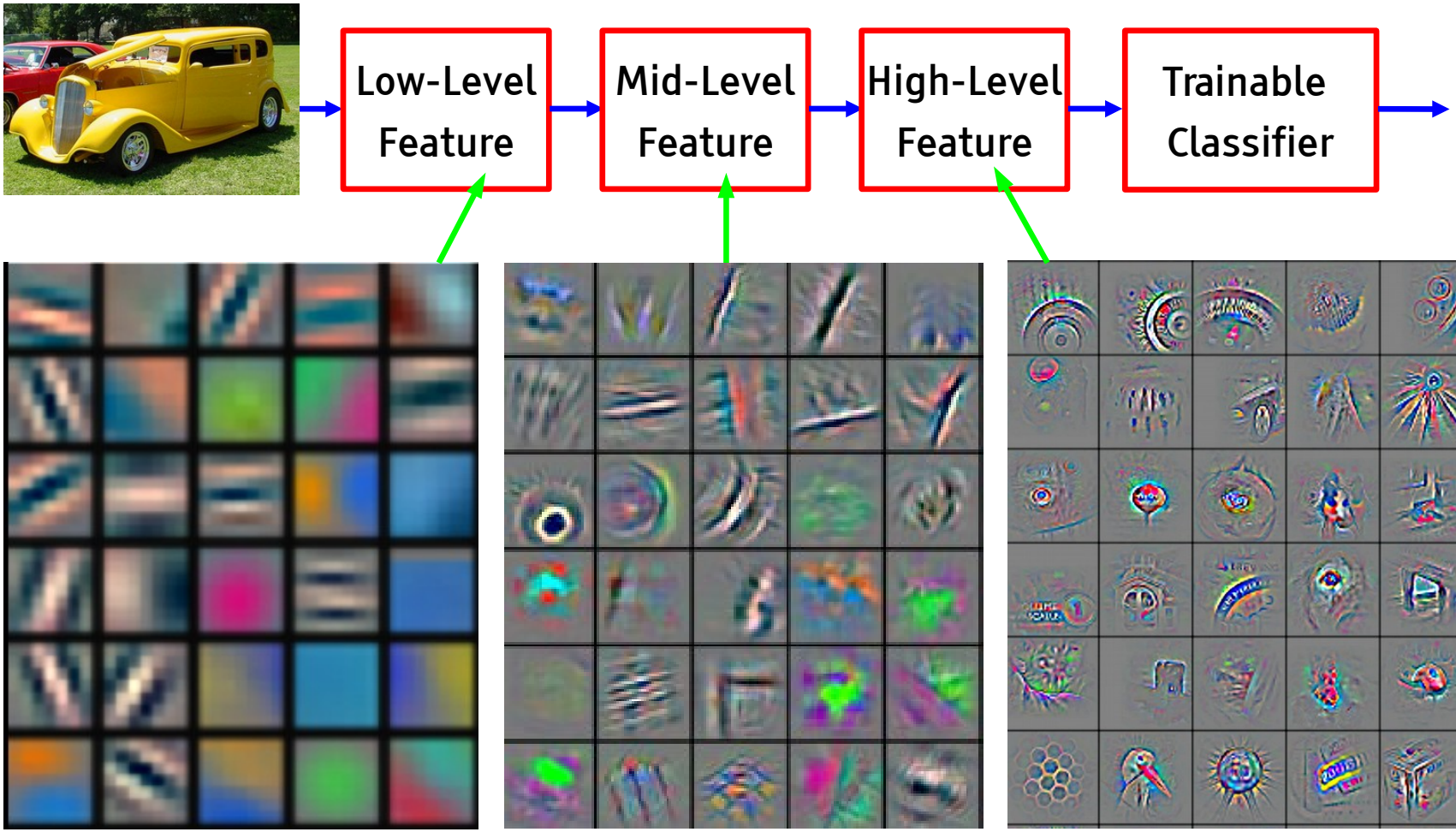
18 August 2015

Aalto University

# Deep Learning = Learning Hierarchical Representations

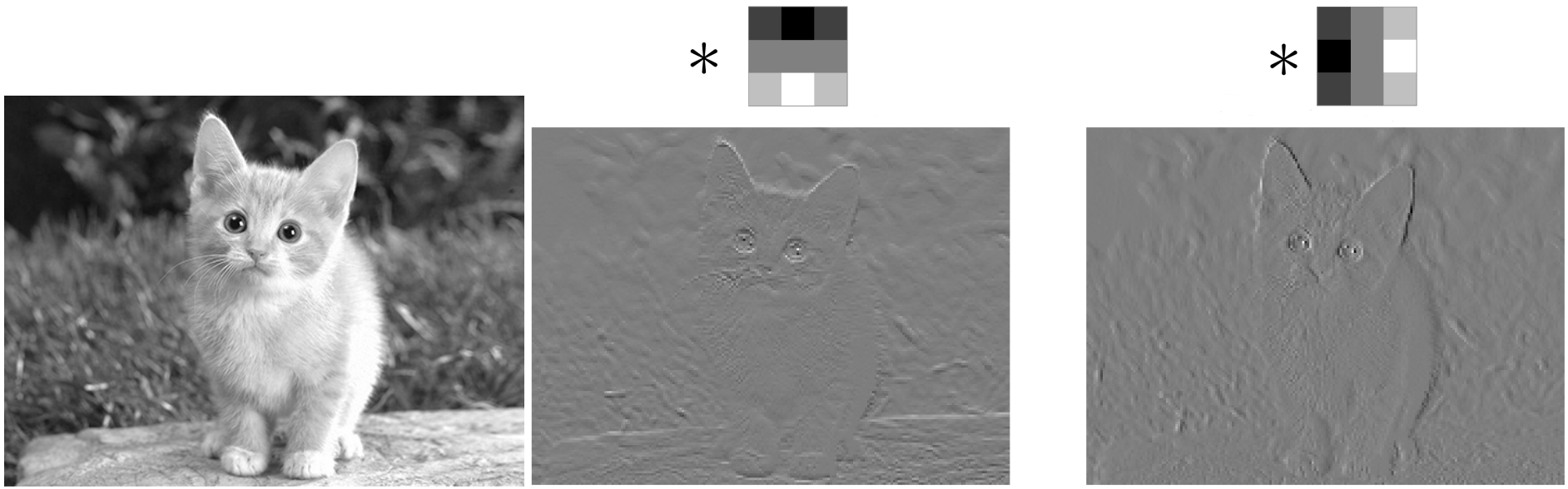
Y LeCun

- 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]

# Convolution $*$



$$(w * x)[t] = \sum_a w[a]x[t-a]$$

Aalto University

# Table of Contents



Aalto University

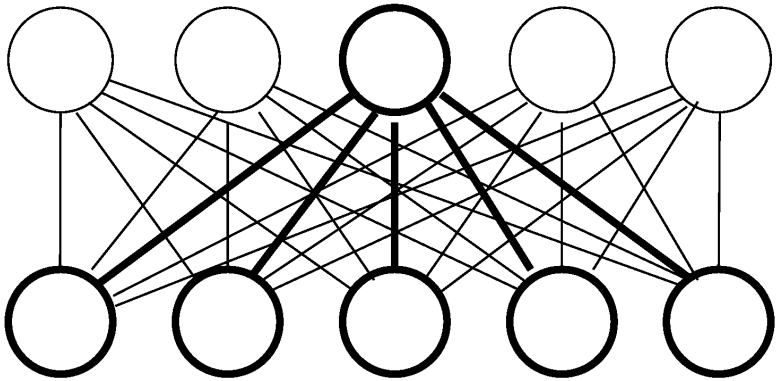
# Convolutional Networks (LeCun et al., 1989)

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

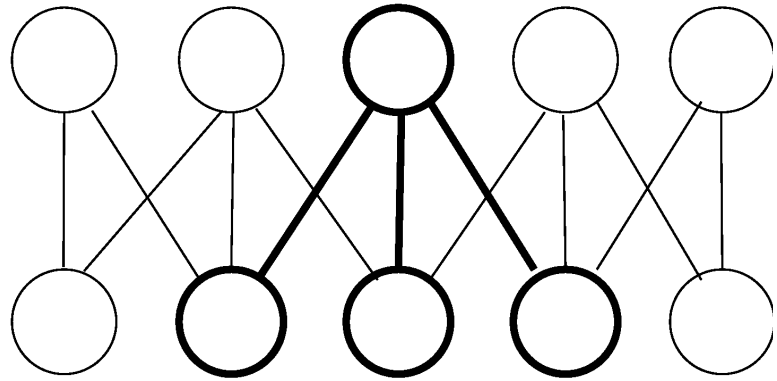
- Change notation: Replace vector signals with tensors indexed by $x, y, c$ for $x, y$ coordinates and channel $c$.
- Weight between $h^{(1)}_{x_1,y_1,c_1}$ and $h^{(2)}_{x_2,y_2,c_2}$ is $W^{(2)}_{x_1,x_2,y_1,y_2,c_1,c_2}$.
- Two ideas: Local connectivity and parameter sharing

Aalto University

# Local connectivity



25 parameters

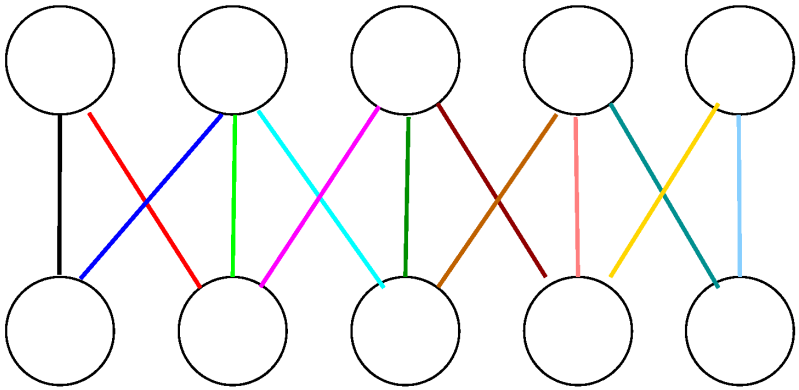


13 parameters

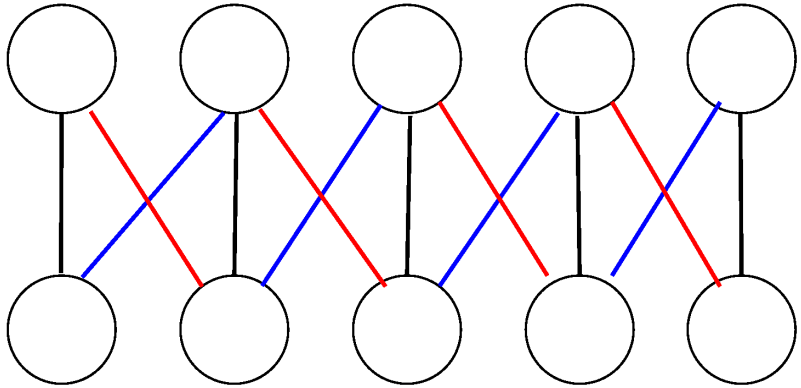
Weight is zero unless $|x_1 - x_2| \leq k$ and $|y_1 - y_2| \leq k$

Aalto University

# Parameter sharing



13 parameters

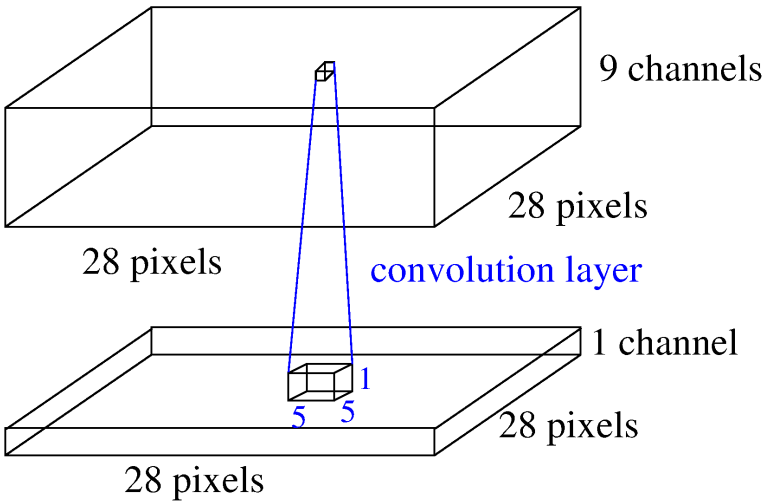


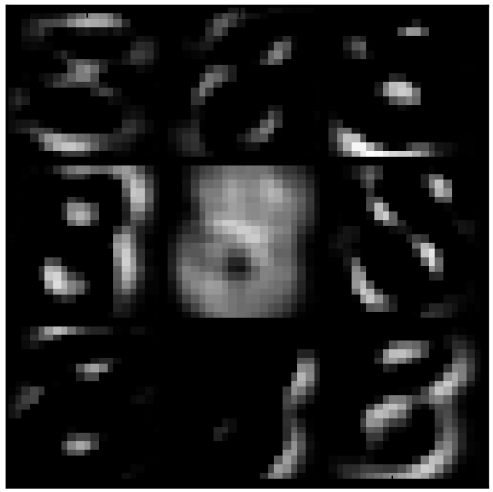
3 parameters

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 $W_{x_1, x_2, y_1, y_2, c_1, c_2}$.

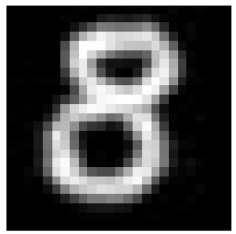
Aalto University

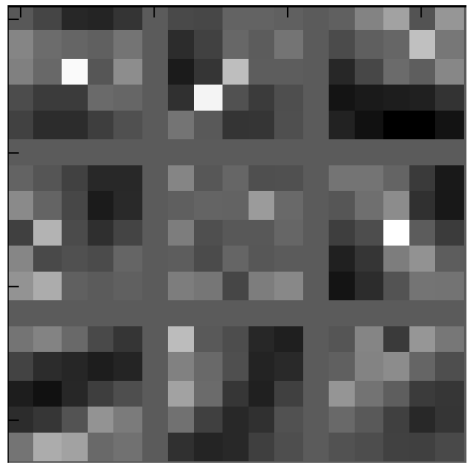
# Example (MNIST again)



signals





weights

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

Aalto University

# Hence Convolutional

$$h_{x,y,c_1} = \text{relu}\left(\sum_{c_0,\Delta x,\Delta y} W_{\Delta x,\Delta y,c_0,c_1} x_{x+\Delta x,y+\Delta y,c_0} + b_{c_1}\right)$$

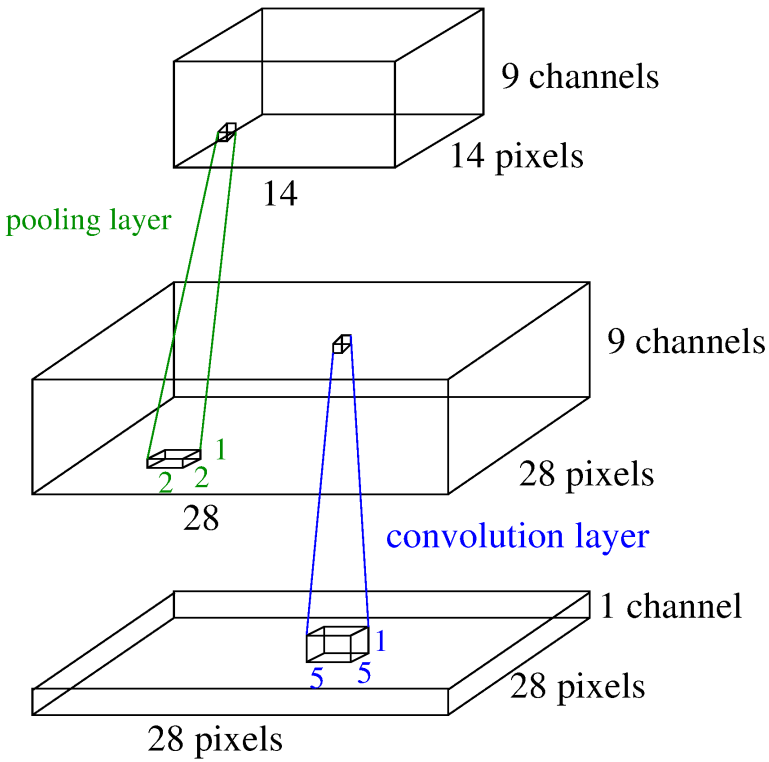
$$\Leftrightarrow \mathbf{h}_{:,:,c_1} = \text{relu}\left(\sum_{c_0} \mathbf{W}_{:,:,c_0,c_1} * \mathbf{x}_{:,:,c_0} + b_{c_1}\right)$$

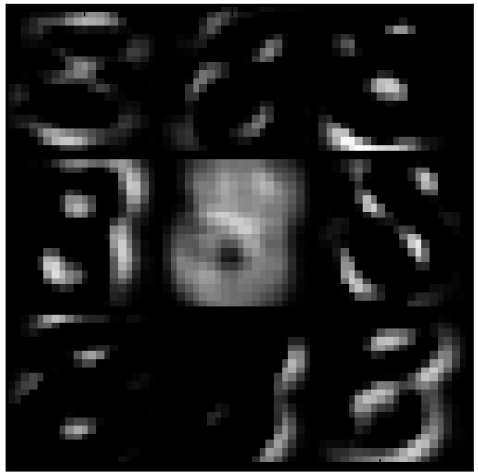
Can be written using the convolution operator $*$.
(Convolution involves flipping one of the two inputs around.
This is not needed, so the actual operation is cross-correlation.)

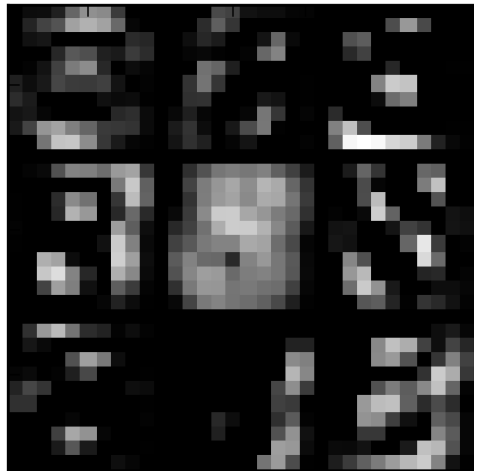
**Padding:** Input can be padded with zeroes to keep the same size.

Aalto University

# Pooling





Decrease resolution and increase channels going up.

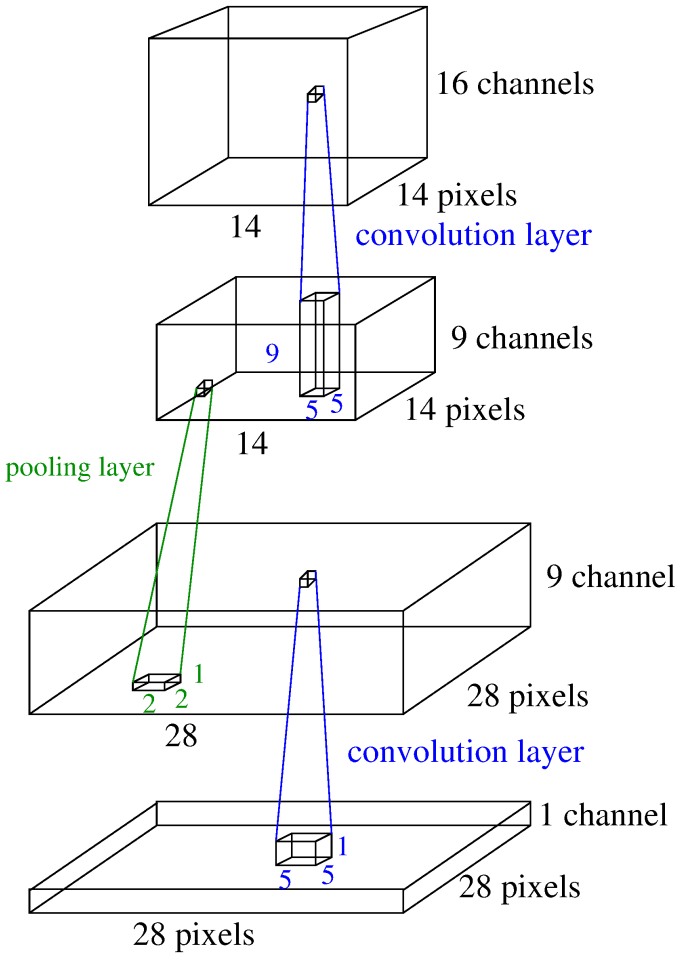
Often down-sampling by hard-coded pooling layers.

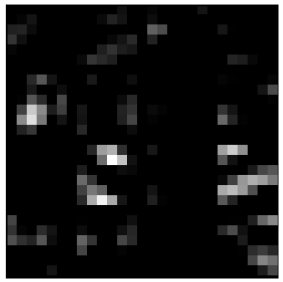
We use $\max(\cdot)$ of $2 \times 2$ activations.

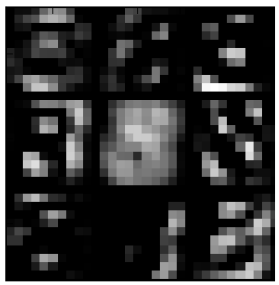
Analysis of pooling functions (Boureau et al., 2010)

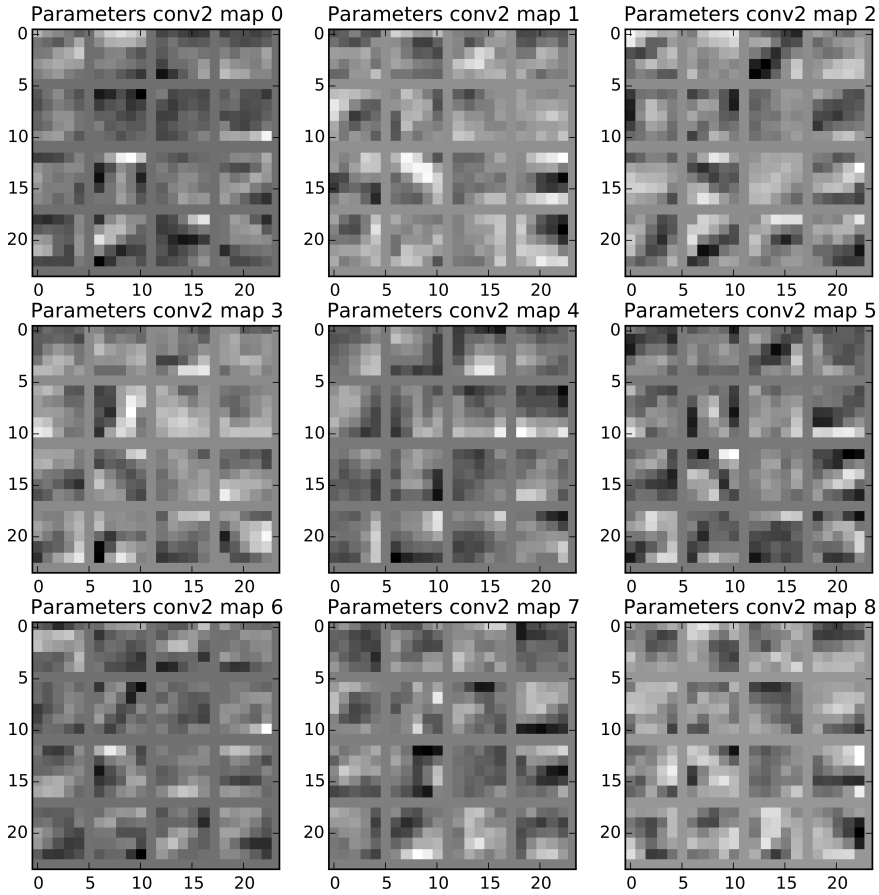
Aalto University

# Stack more layers



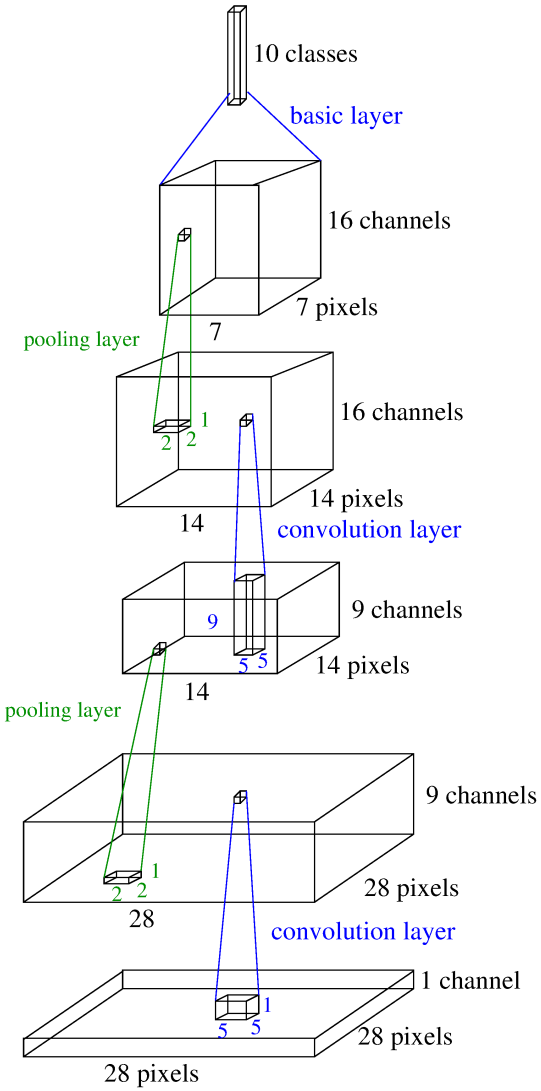


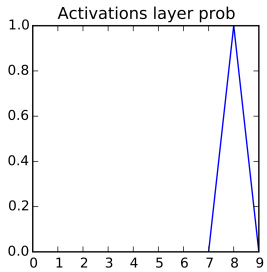


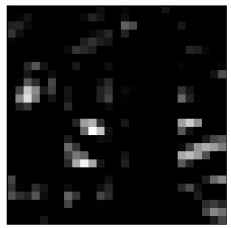

Note how each unit looks at all channels of the previous layer.

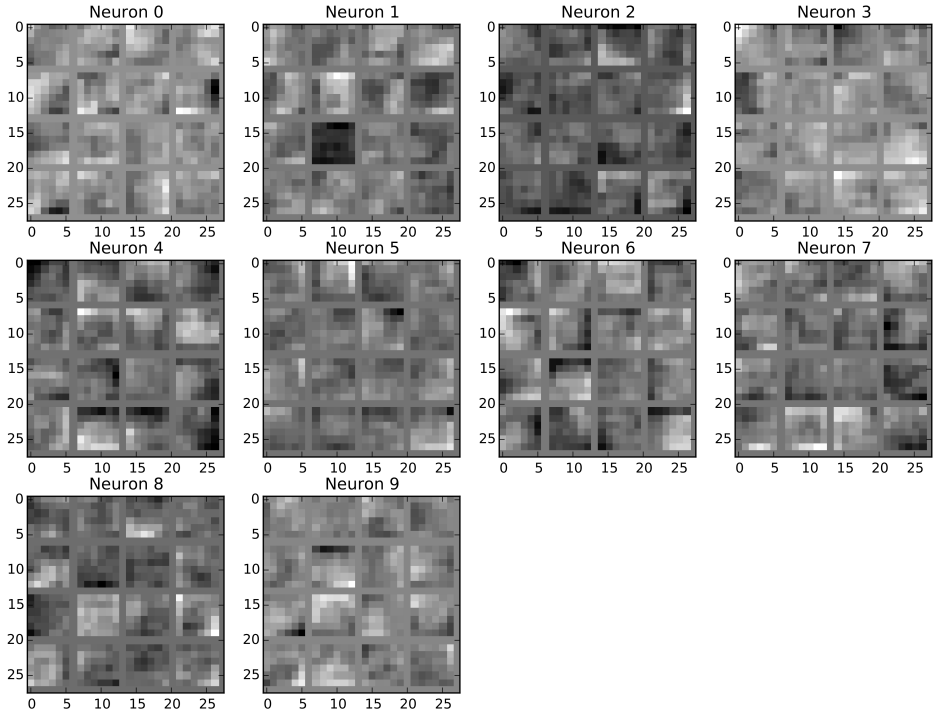
Aalto University

# Whole network



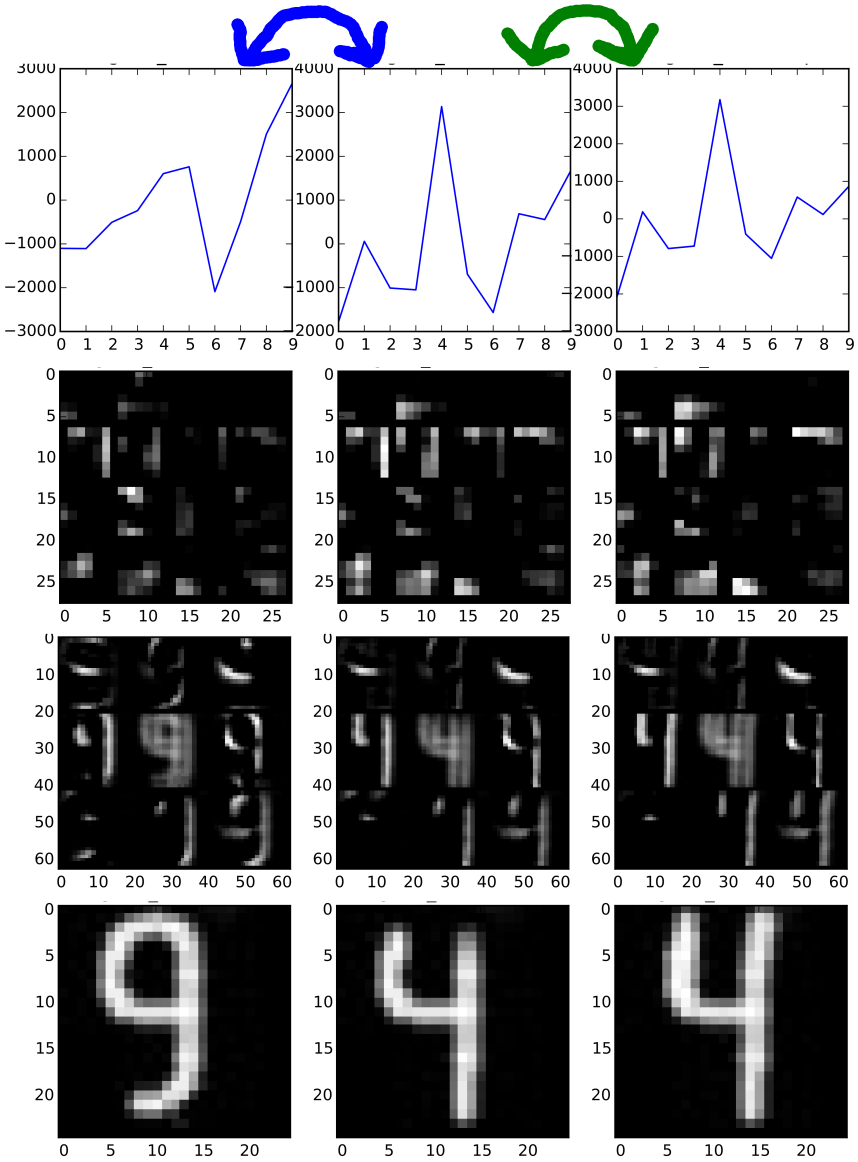


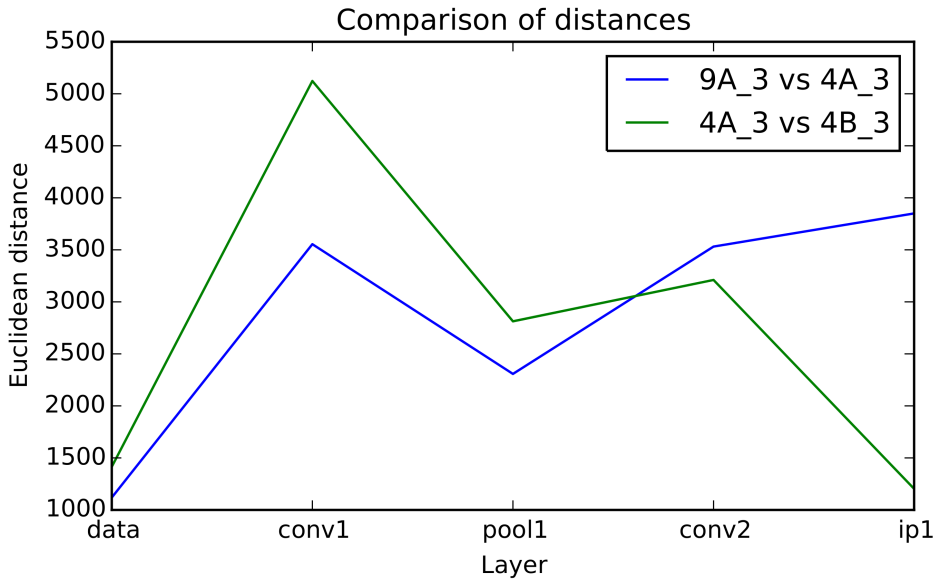

At the top, the resolution drops to $1 \times 1$ pixel.

Train by back-propagation as before.

Aalto University

# Good representation?






In pixel space, 9–4 is closer.
In feature space, 4–4 is closer.

Aalto University

# Convolutional vs. non-convolutional

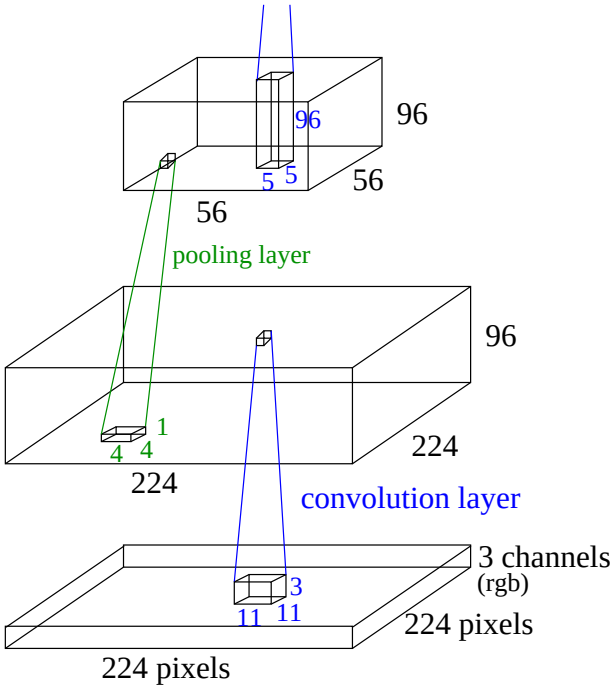
Number of weights (ignoring biases):
$5{\cdot}5{\cdot}9{+}5{\cdot}5{\cdot}9{\cdot}16{+}7{\cdot}7{\cdot}16{\cdot}10 = 225{+}3600{+}7840 = 11665$
Sizes of signals $\mathbf{h}$:
$28 \cdot 28, 28 \cdot 28 \cdot 9, 14 \cdot 14 \cdot 16 = 784, 7056, 3136$

Compare to the introductory MNIST 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.

Aalto University

# Scaling up



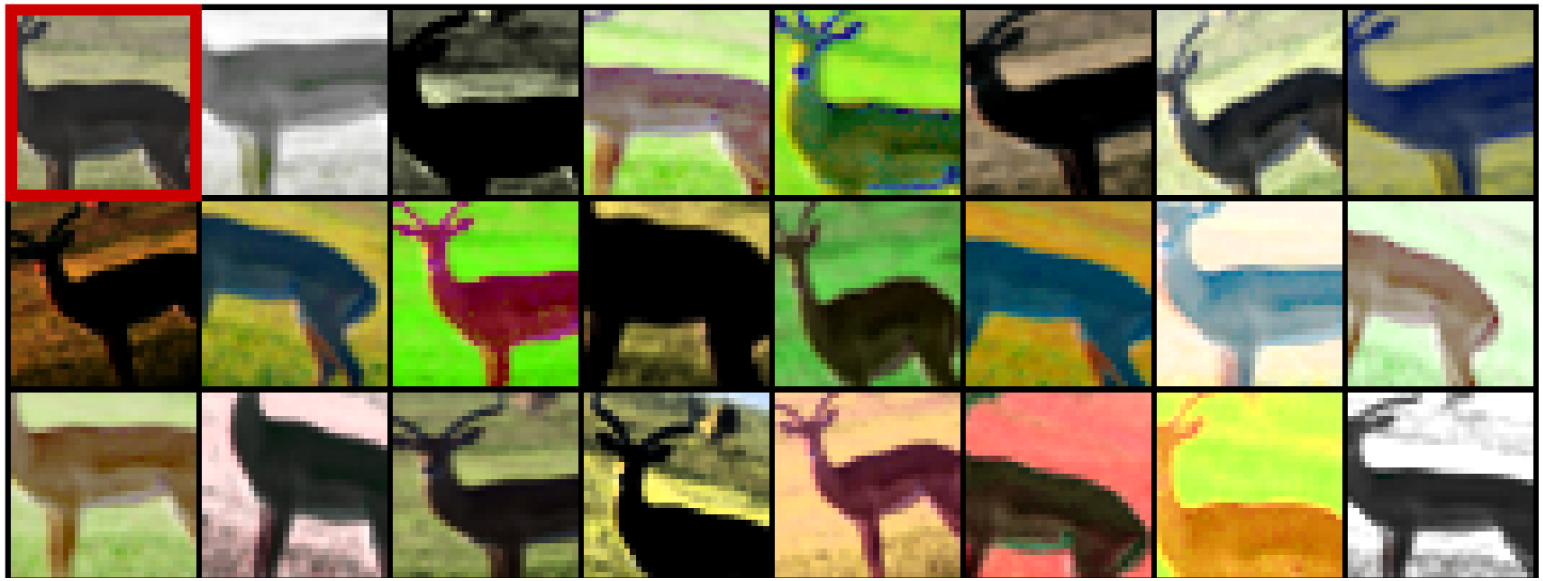

First layers of (Krizhevsky, 2012).
Trend in 2015 towards small filters ($3 \times 3$) and poolings ($2 \times 2$), but lots of layers (10–40).

`github.com/BVLC/caffe/blob/master/examples/00-classification.ipynb`

Aalto University

# Regularization by data augmentation

Image from (Dosovitskiy et al., 2014)

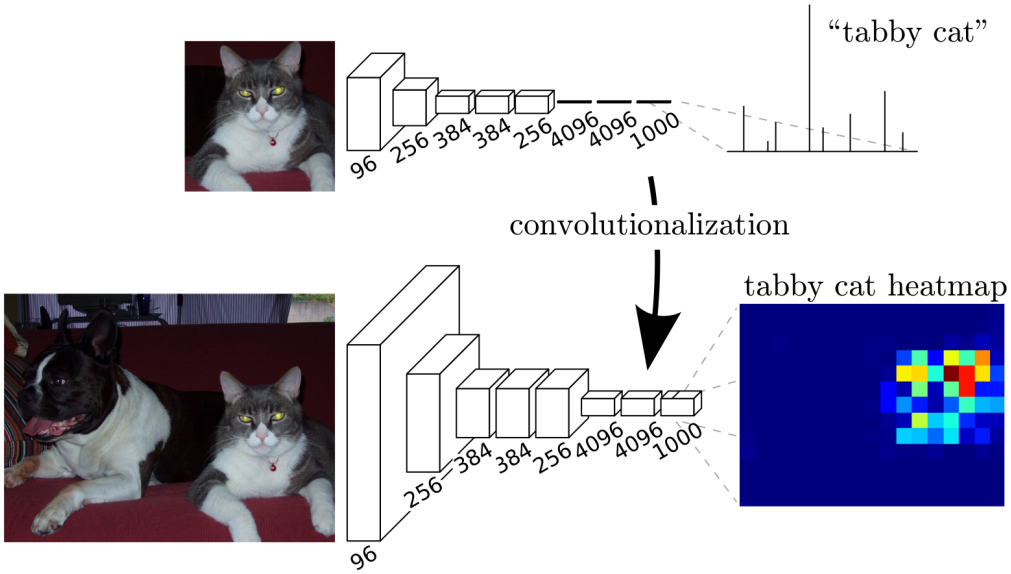


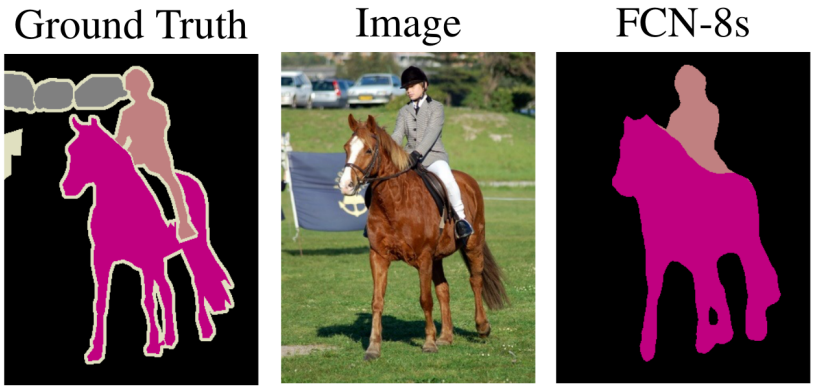
Augmented data by image-specific transformations.
E.g. cropping just 2 pixels gets you 9 times the data!

Infinite MNIST: `http://leon.bottou.org/projects/infimnist`

Aalto University

# Semantic segmentation (Long et al., 2015)



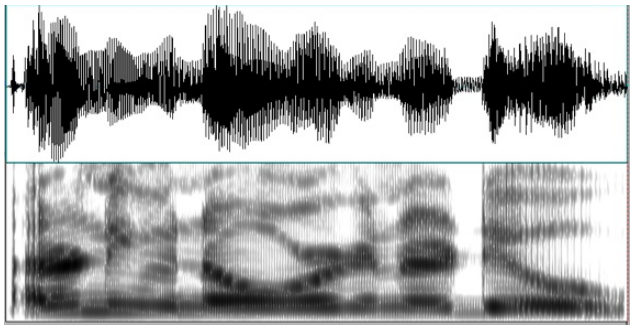



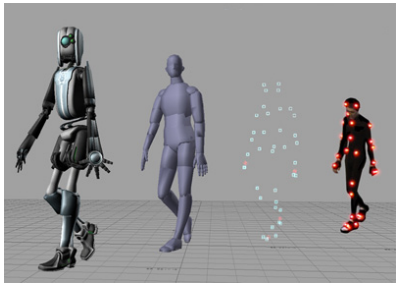

Sliding a convolutional network to classify each location.
Lots of shared computation.

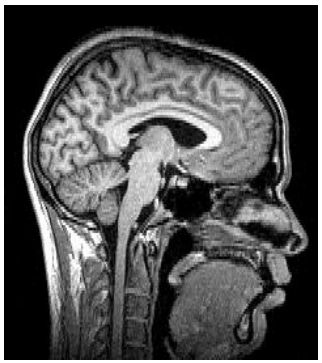
Aalto University

# 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 |







Aalto University

# Table of Contents



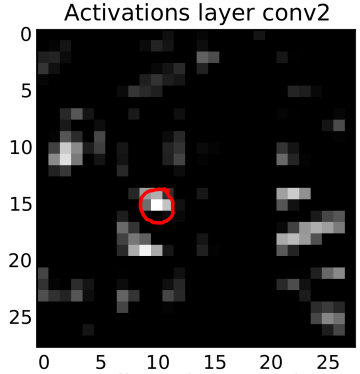
Aalto University

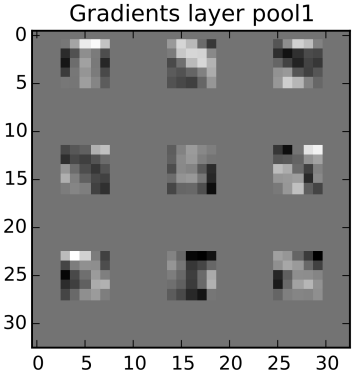
# Gradients to inputs (Zeiler and Fergus, 2013)

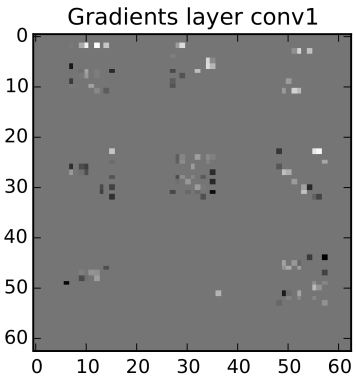
- Let us examine the network by investigating gradients.
- Usually, gradients are propagated from training criterion $C$ to parameters $\boldsymbol{\theta} = \{\mathbf{W}^{(L)}, \mathbf{b}^{(L)}\}$.
- Now, we propagate gradients from hidden units $\mathbf{h}$ to inputs $\mathbf{x}$ for visualization.
- $\frac{\partial \mathbf{h}^{(1)}}{\partial \mathbf{x}}$ is boring, gives just filters $\mathbf{W}^{(1)}$:

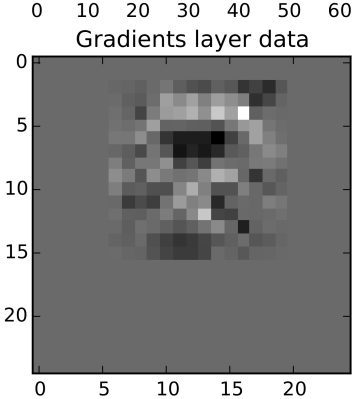
$$\frac{\partial \mathbf{h}^{(1)}}{\partial \mathbf{x}} = \frac{\partial}{\partial \mathbf{x}} \text{relu}\left(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}\right) = \mathbf{1}\left(\mathbf{h}^{(1)} > 0\right)\mathbf{W}^{(1)}$$

Aalto University

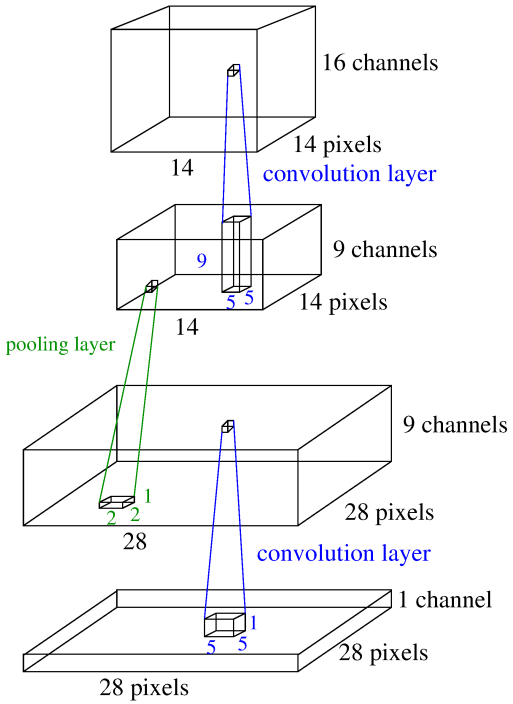


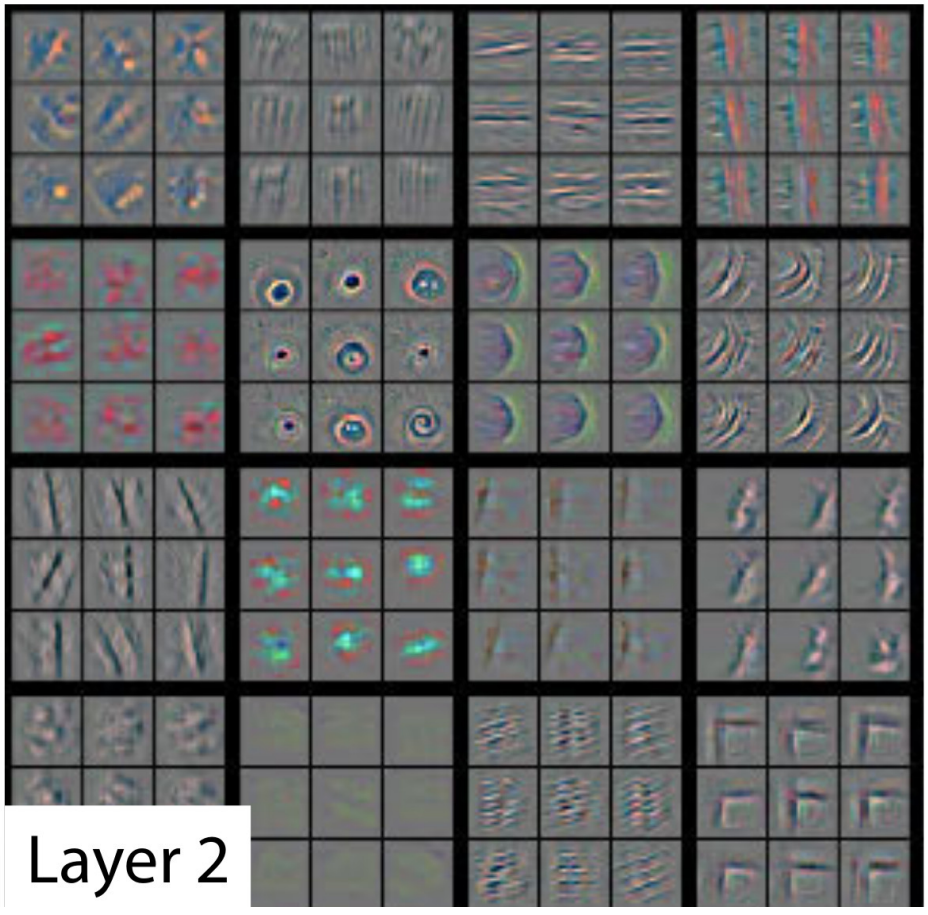



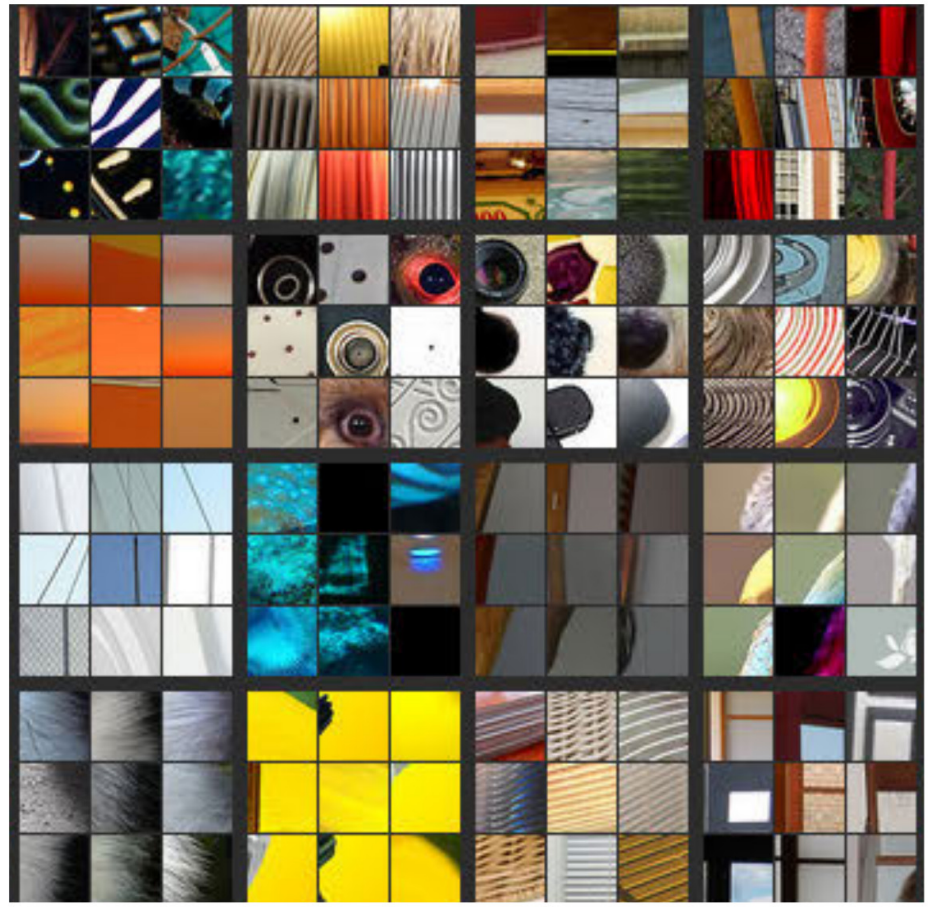



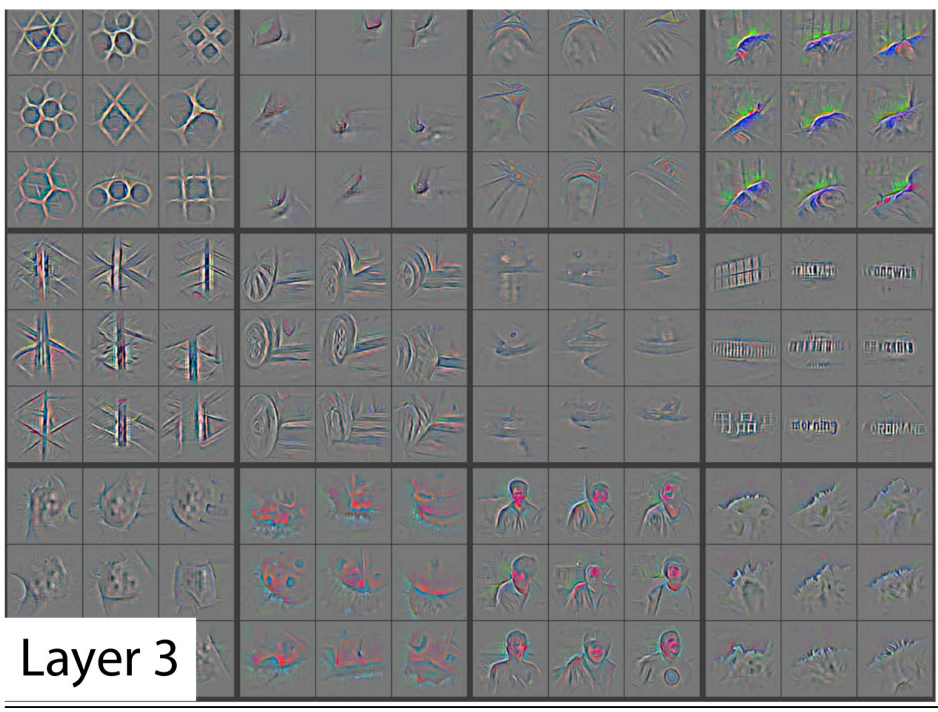

- Pick most active unit $h^{(2)}_{x,y,c}$.
- See filters of $\mathbf{W}^{(2)}$.
- Max-pooling and relu,
  only some signals get through.
- In input space $(\partial h^{(2)}_{x,y,c}/\partial \mathbf{x})$, you see it was paying attention to the upper loop of digit 8.



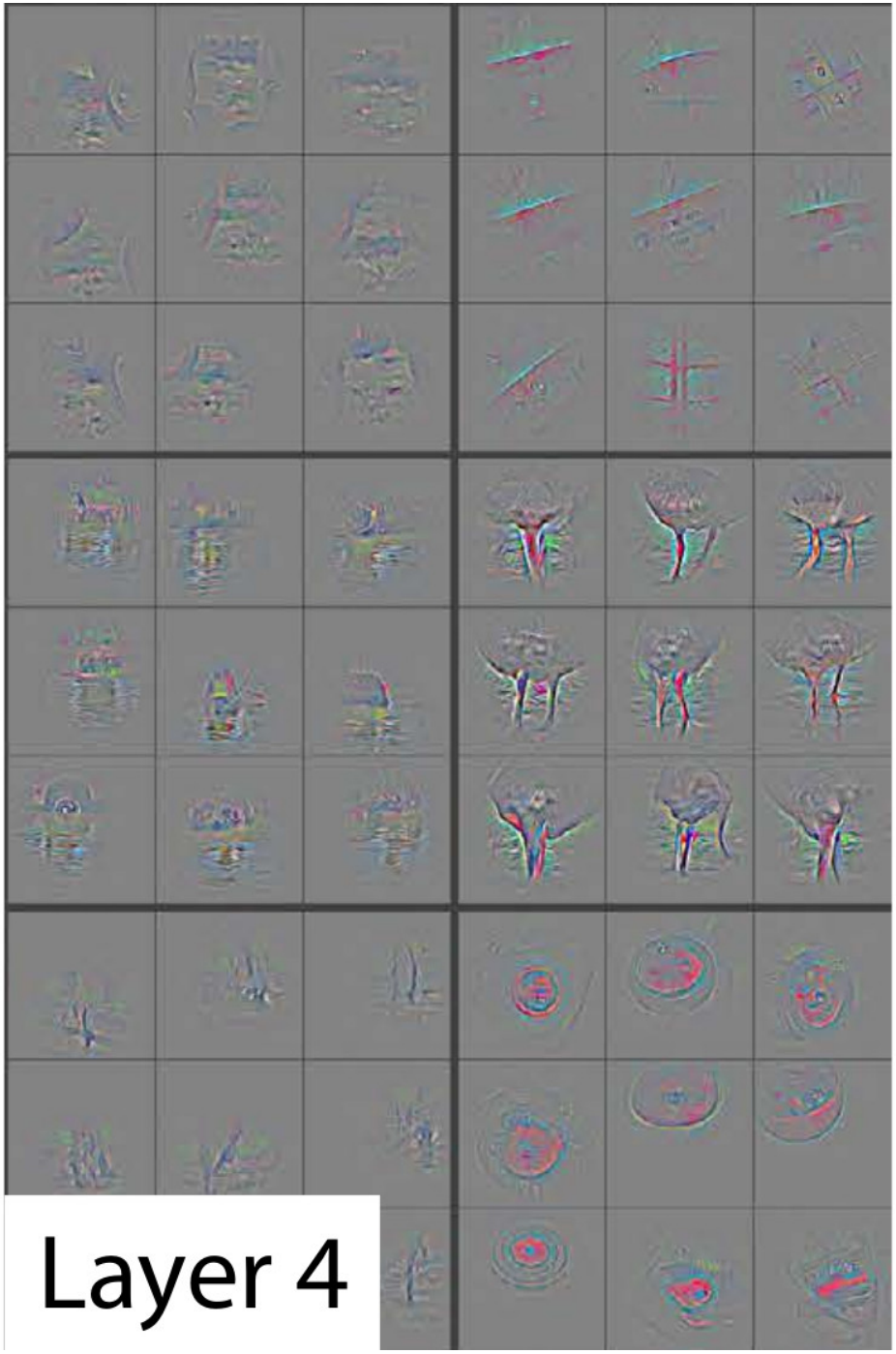


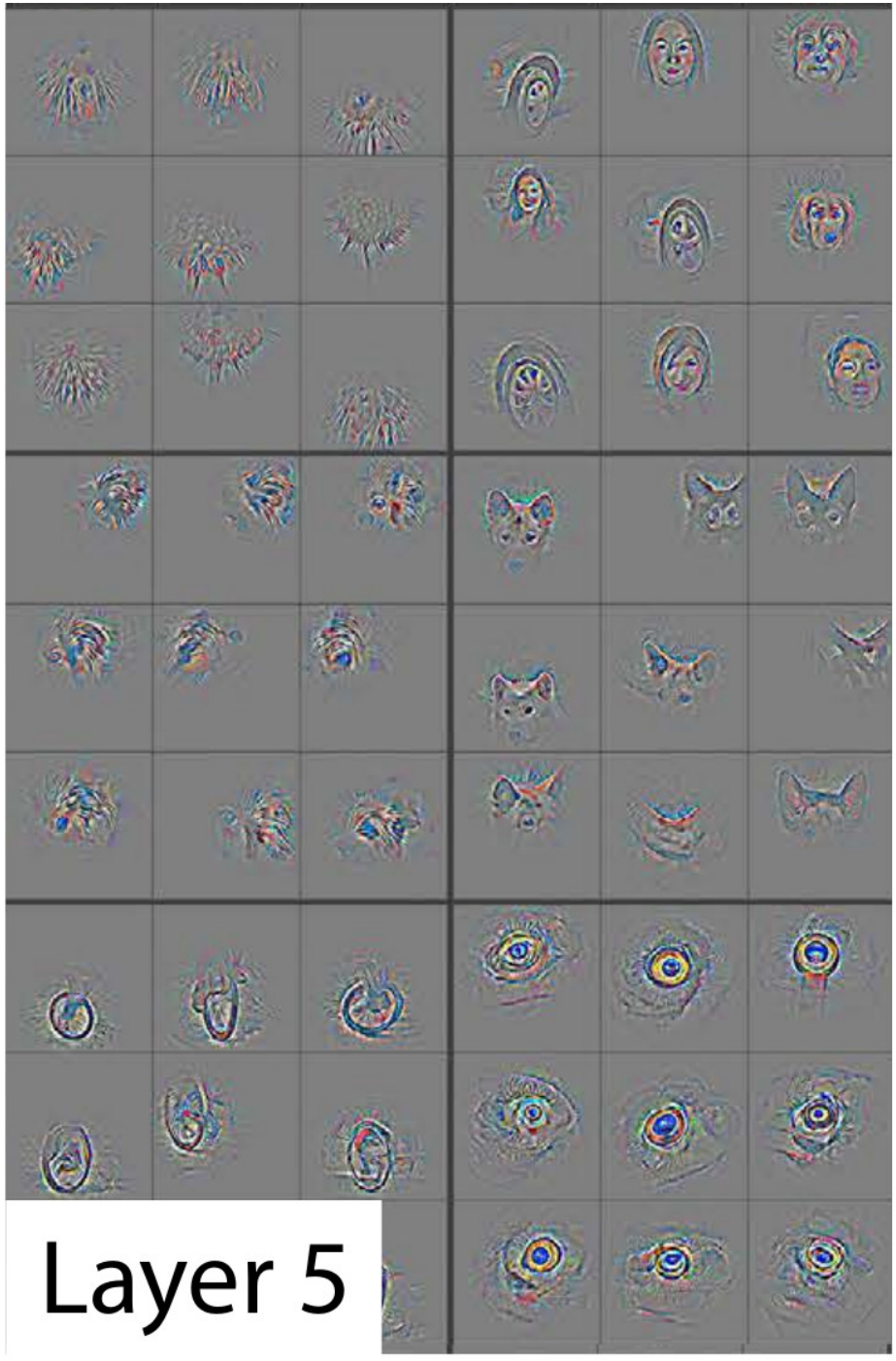
Bigger network, find examples that activate a unit most, work backwards as before. (Zeiler and Fergus, 2013)

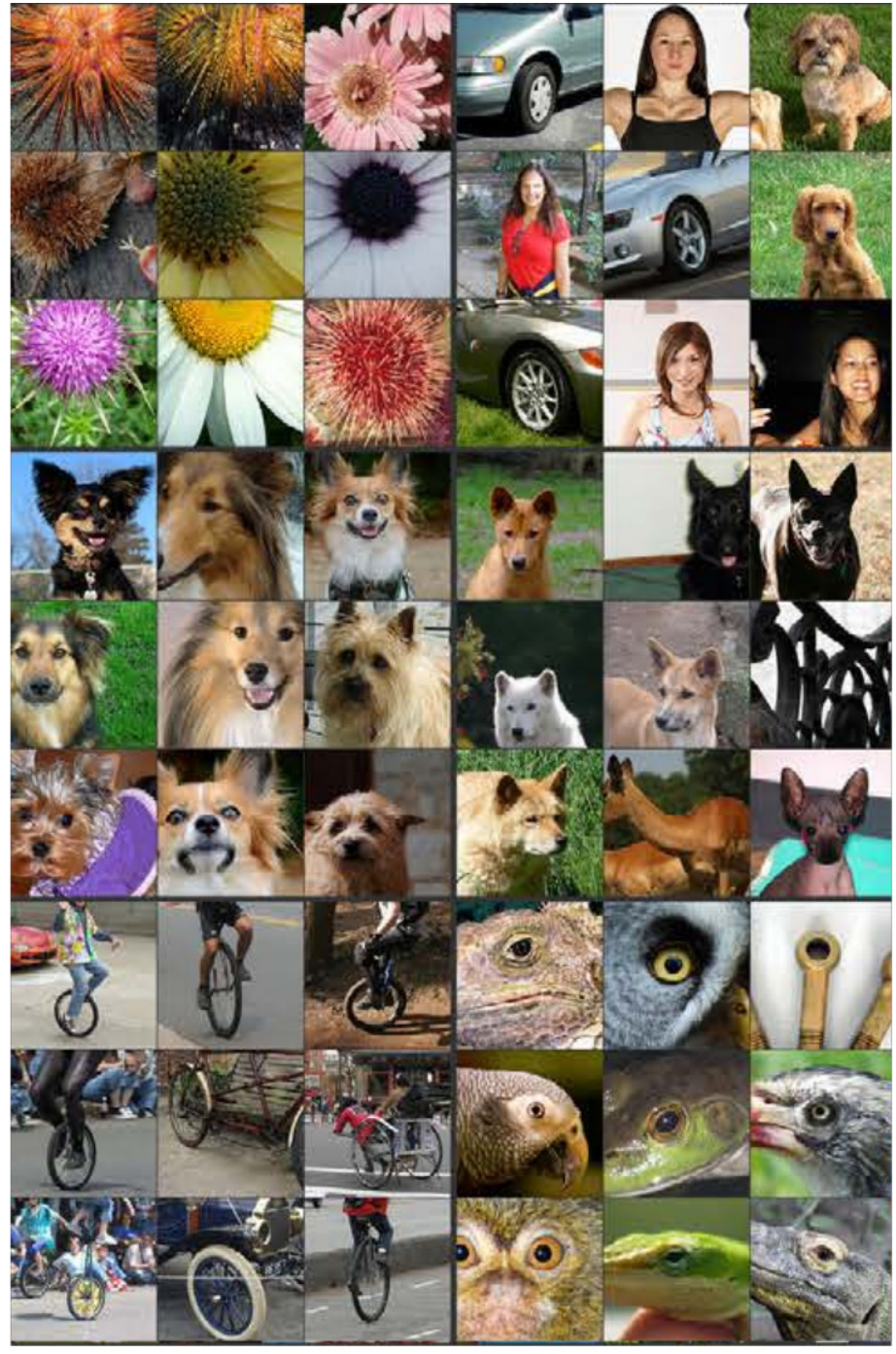

Layer 3

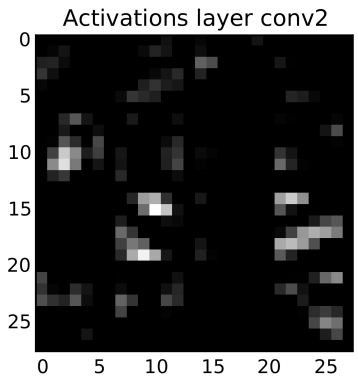

STRIKE FACE
Strongwish
morning

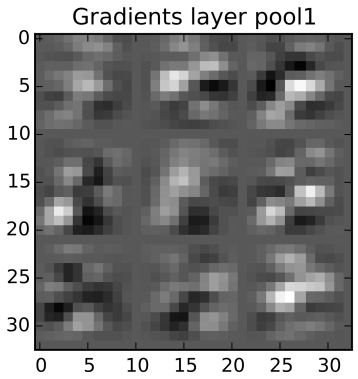


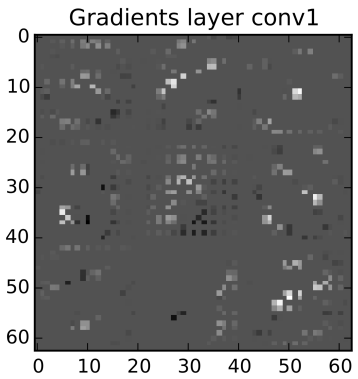
Layer 4

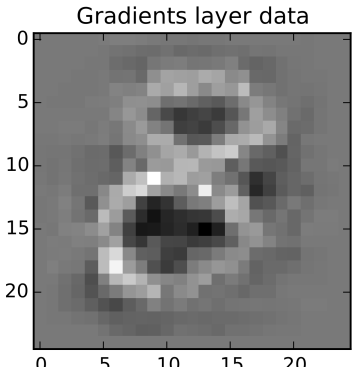

Layer 5





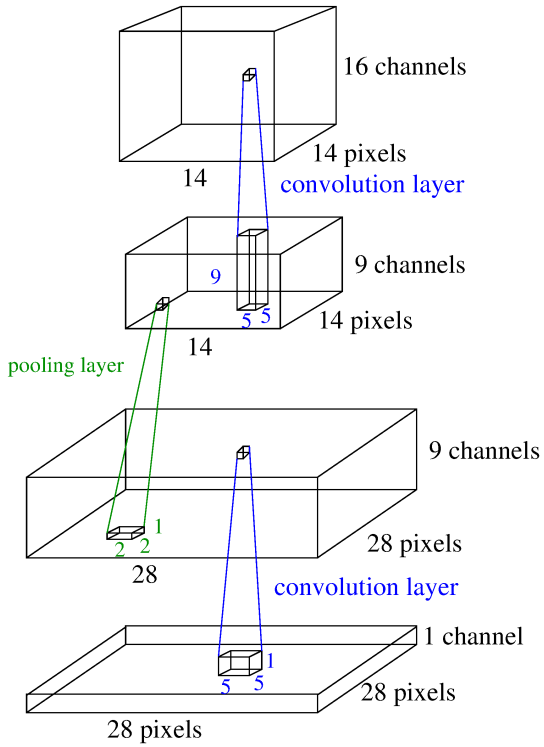



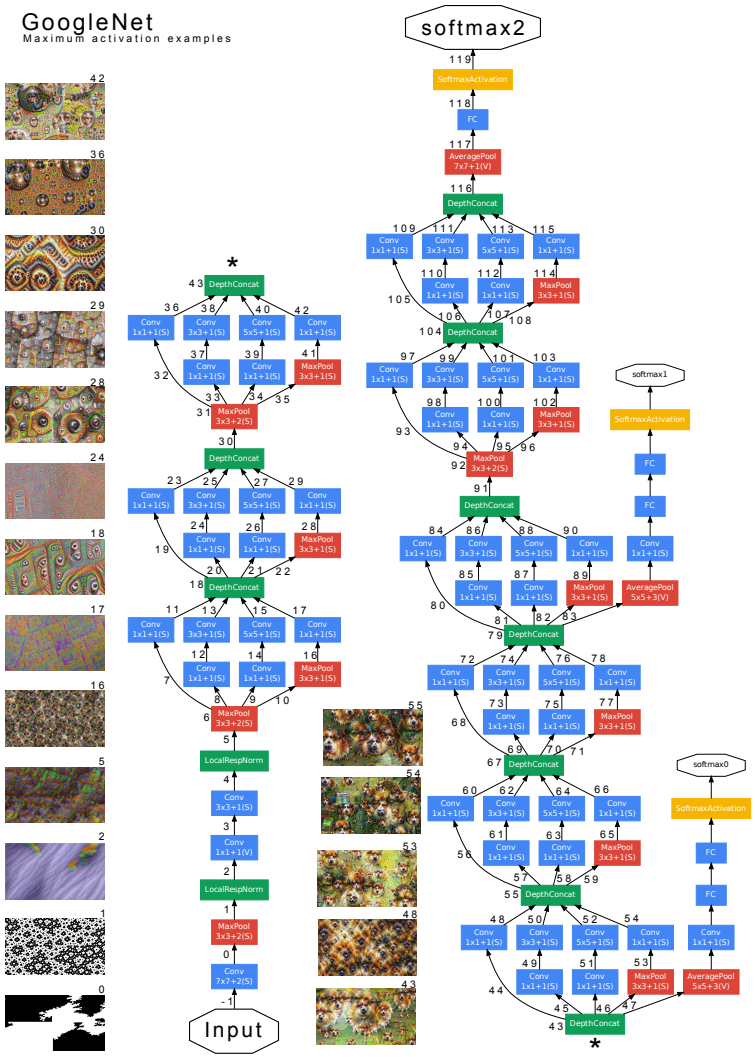

# What would activate the whole layer more?

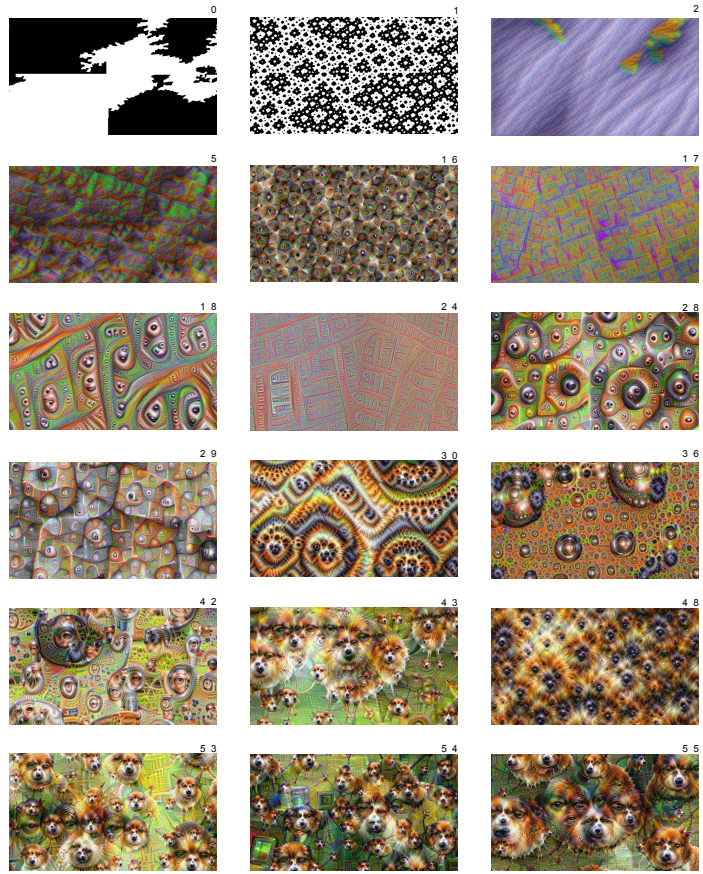
$$\frac{\partial \|\mathbf{h}\|^2}{\partial \mathbf{x}} = 2\mathbf{h}^\top \frac{\partial \mathbf{h}}{\partial \mathbf{x}}$$




Generate hallucinations: Change input a bit towards what is "seen".

# Inceptionism (Mordvintsev, 2015) Video by Miquel Perello Nieto



youtube.com/watch?v=w5U7EL72ngI users.ics.aalto.fi/perellm1/deep_dreams.shtml

Aalto University

# Thanks for listening!

Thanks to Miquel Perello Nieto for examples.

Aalto University

# Possible Exercises

- Follow the Theano tutorial on convolutional networks: `deeplearning.net/tutorial/lenet.html`
- Caffe code used for the slides: `github.com/perellonieto/cnn_visualization_caffe`

Aalto University