6. Convolutional Neural Networks

CS 535 Deep Learning, Winter 2018

Fuxin Li

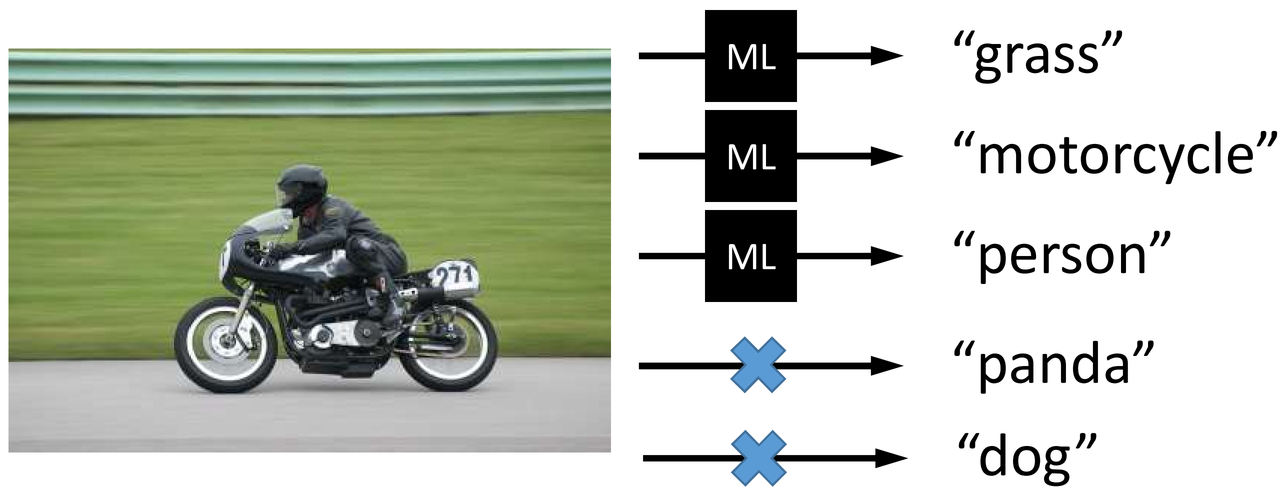
With materials from Zsolt Kira

Quiz coming up…

- Next Monday (2/5)
- 30 minutes
- Topics:
  - Optimization
  - Basic neural networks
  - Neural Network Optimization
- No Convolutional nets in this quiz
- No “Theoretical Implications” part
  - e.g. topics such as Assignment 1 question 1, initial quiz questions concerning high-dimensional space, etc. won’t be covered in the quiz

The Image Classification Problem

(Multi-label in principle)



Neural Networks

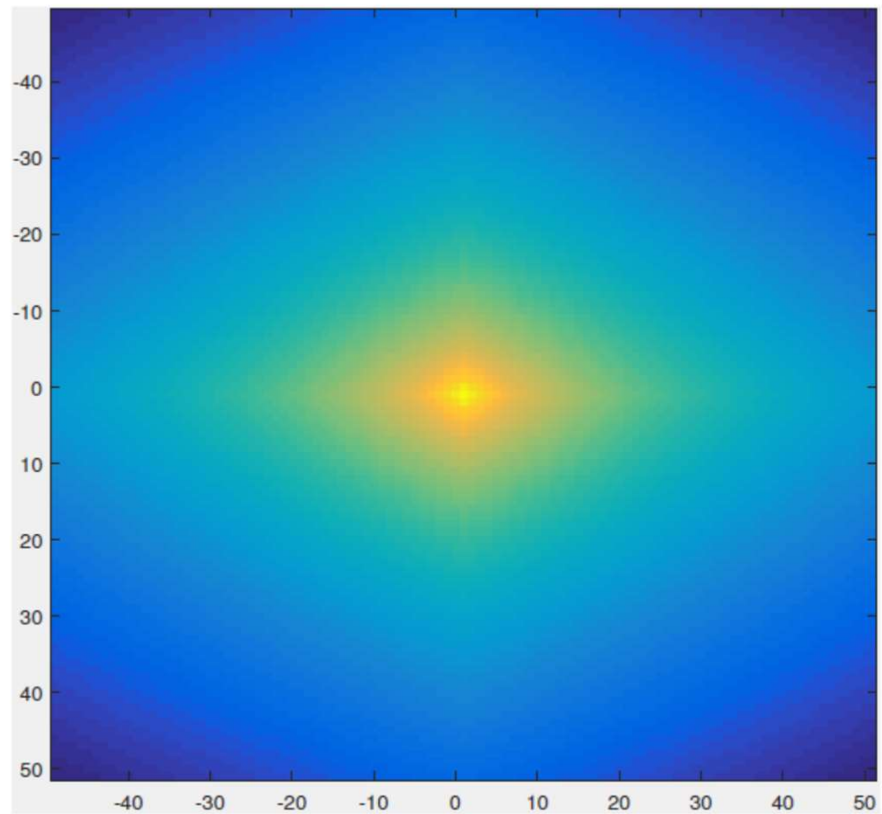
- Extremely high dimensionality!
- 256x256 image has already 65,536 * 3 dimensions
- One hidden layer with 500 hidden units require 65,536 * 3 * 500 connections (98 Million parameters)

Challenges in Image Classification

Structure between neighboring pixels in natural images

Takeaways:
1) Long-range correlation
2) Local correlation stronger than non-local

The correlation prior for horizontal and vertical shifts (averaged over 1000 images) looks like this:



The convolution operator

Sobel filter

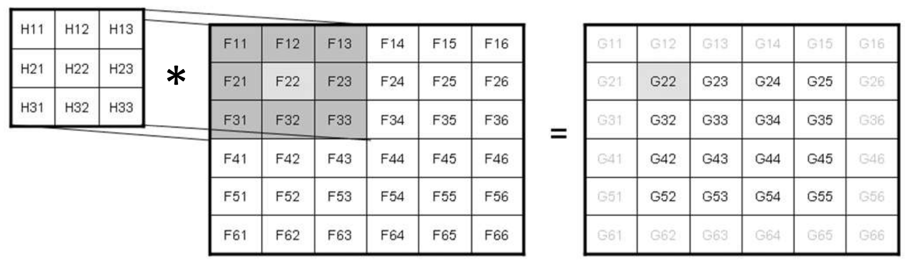
-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Convolution



I

Convolution

$I * Gx$

2D Convolution with Padding

0	0	0	0	0
0	1	3	1	0
0	0	-1	1	0
0	2	2	-1	0
0	0	0	0	0

*

-2	-2	1
-2	0	1
1	1	1

=

2D Convolution with Padding

0	0	0	0	0
0	1	3	1	0
0	0	-1	1	0
0	2	2	-1	0
0	0	0	0	0

*

-2	-2	1
-2	0	1
1	1	1

=

2		

$$3 \times 1 + (-1) \times 1 = 2$$

2D Convolution with Padding

0	0	0	0	0
0	1	3	1	0
0	0	-1	1	0
0	2	2	-1	0
0	0	0	0	0

$*$

-2	-2	1
-2	0	1
1	1	1

$=$

2	-1	

$$1 \times (-2) + 1 \times 1 + 1 \times (-1) + 1 \times 1 = -1$$

2D Convolution with Padding

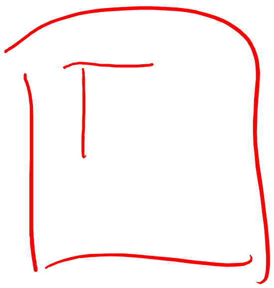
0	0	0	0	0
0	1	3	1	0
0	0	-1	1	0
0	2	2	-1	0
0	0	0	0	0

*

-2	-2	1
-2	0	1
1	1	1

=

2	-1	-6



What if:

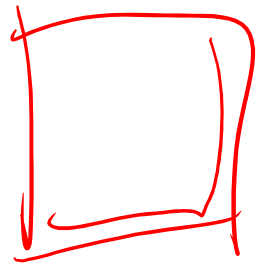
0	0	3	3	0
0	1	3	1	0
0	0	-1	1	0
0	2	2	-1	0
0	0	0	0	0

*

-2	-2	1
-2	0	1
1	1	1

=

2	-1	-18



2D Convolution with Padding

0	0	0	0	0
0	1	3	1	0
0	0	-1	1	0
0	2	2	-1	0
0	0	0	0	0

*

-2	-2	1
-2	0	1
1	1	1

=

2	-1	-6
4		

2D Convolution with Padding

0	0	0	0	0
0	1	3	1	0
0	0	-1	1	0
0	2	2	-1	0
0	0	0	0	0

*

-2	-2	1
-2	0	1
1	1	1

=

2	-1	-6
4	-3	

2D Convolution with Padding

0	0	0	0	0
0	1	3	1	0
0	0	-1	1	0
0	2	2	-1	0
0	0	0	0	0

*

-2	-2	1
-2	0	1
1	1	1

=

2	-1	-6
4	-3	-5

2D Convolution with Padding

0	0	0	0	0
0	1	3	1	0
0	0	-1	1	0
0	2	2	-1	0
0	0	0	0	0

*

-2	-2	1
-2	0	1
1	1	1

=

2	-1	-6
4	-3	-5
1		

$\mathrm{ReLU}(w * x + b)$

if $b = -2$

$\mathrm{ReLU}(x) = \max(x, 0)$

2D Convolution with Padding

0	0	0	0	0
0	1	3	1	0
0	0	-1	1	0
0	2	2	-1	0
0	0	0	0	0

*

-2	-2	1
-2	0	1
1	1	1

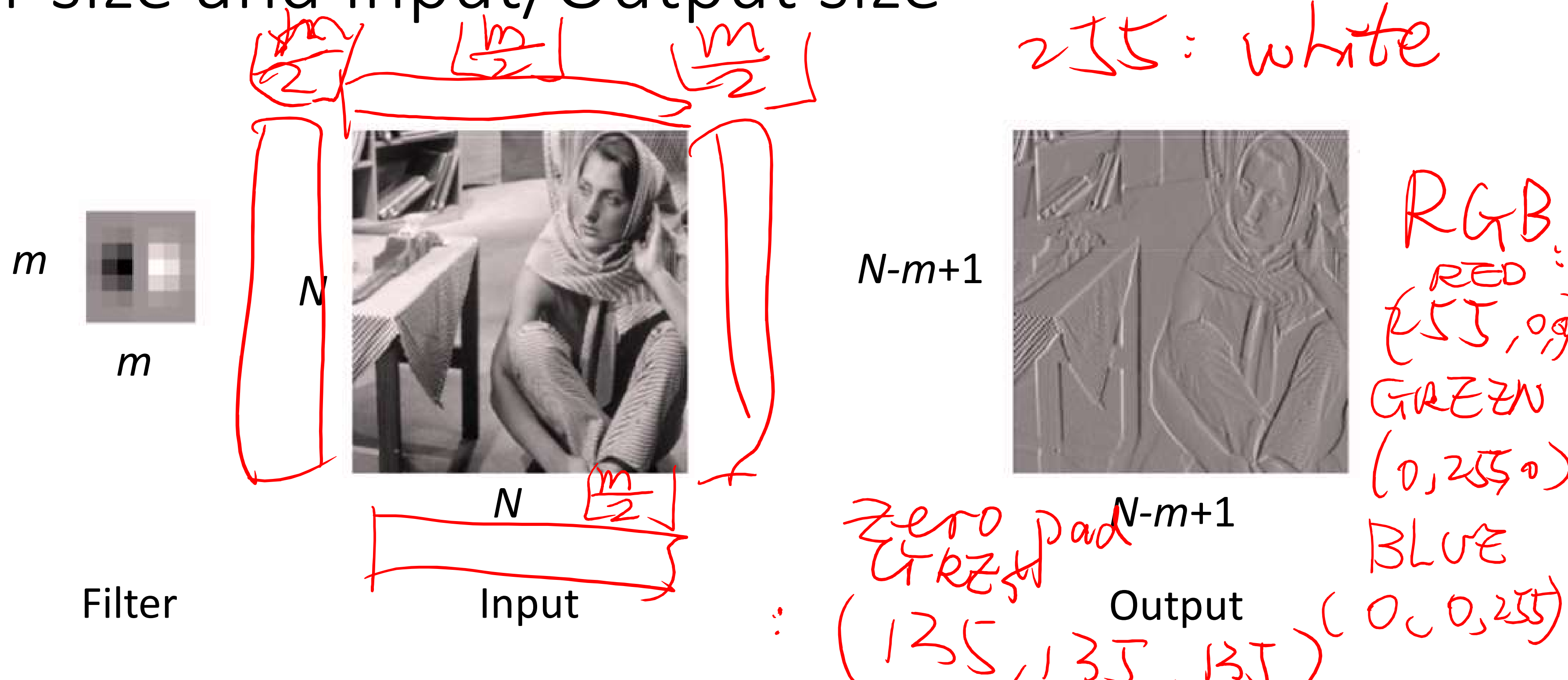
=

2	-1	-6
4	-3	-5
1	-2	

2D Convolution with Padding

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 2 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} -2 & -2 & 1 \\ -2 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & -1 & -6 \\ 4 & -3 & -5 \\ 1 & -2 & -2 \end{bmatrix}$$

Filter size and Input/Output size



- Zero padding the input so that the output is NxN

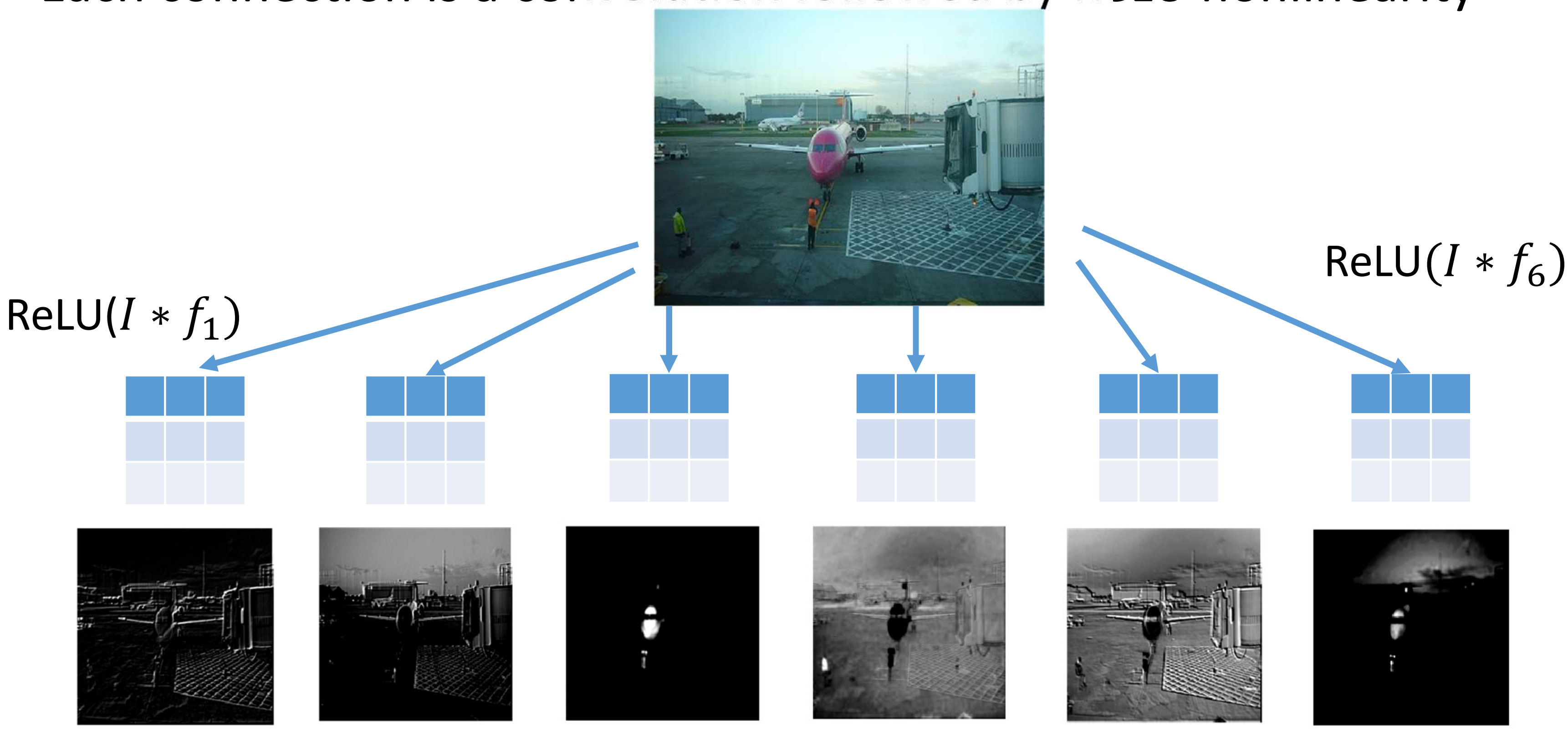
(On the image – mean)

Location-invariance in images

- Image Classification
  - It does not matter where the object appears
- Object Localization
  - It does matter where the object appears
    - (Deconvolution – to be dealt with later)
  - But the rules for recognizing object are the same everywhere in the image

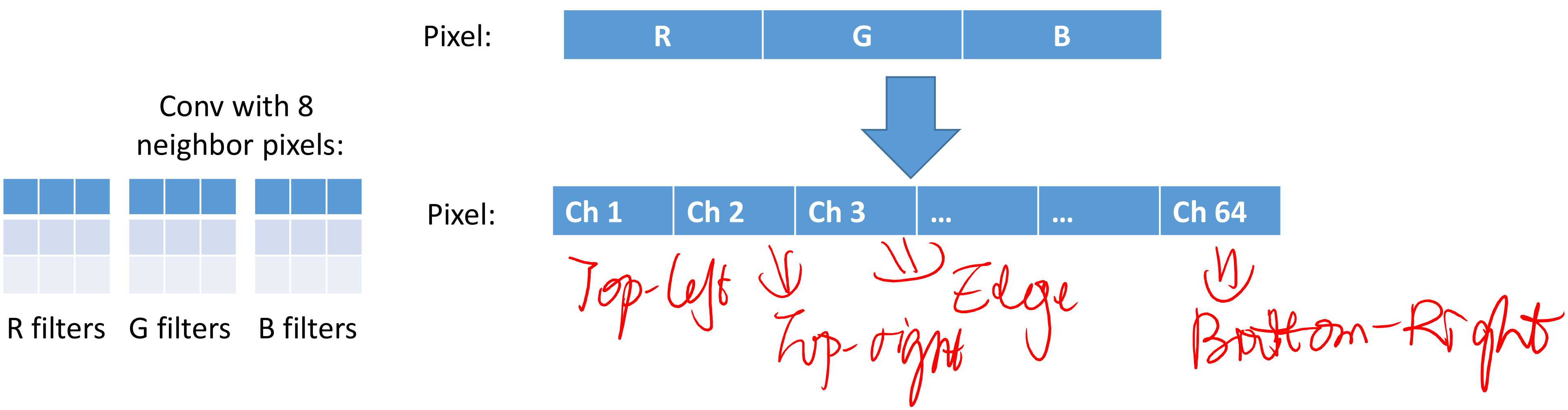
Convolutional Networks

- Each connection is a convolution followed by ReLU nonlinearity



For each pixel

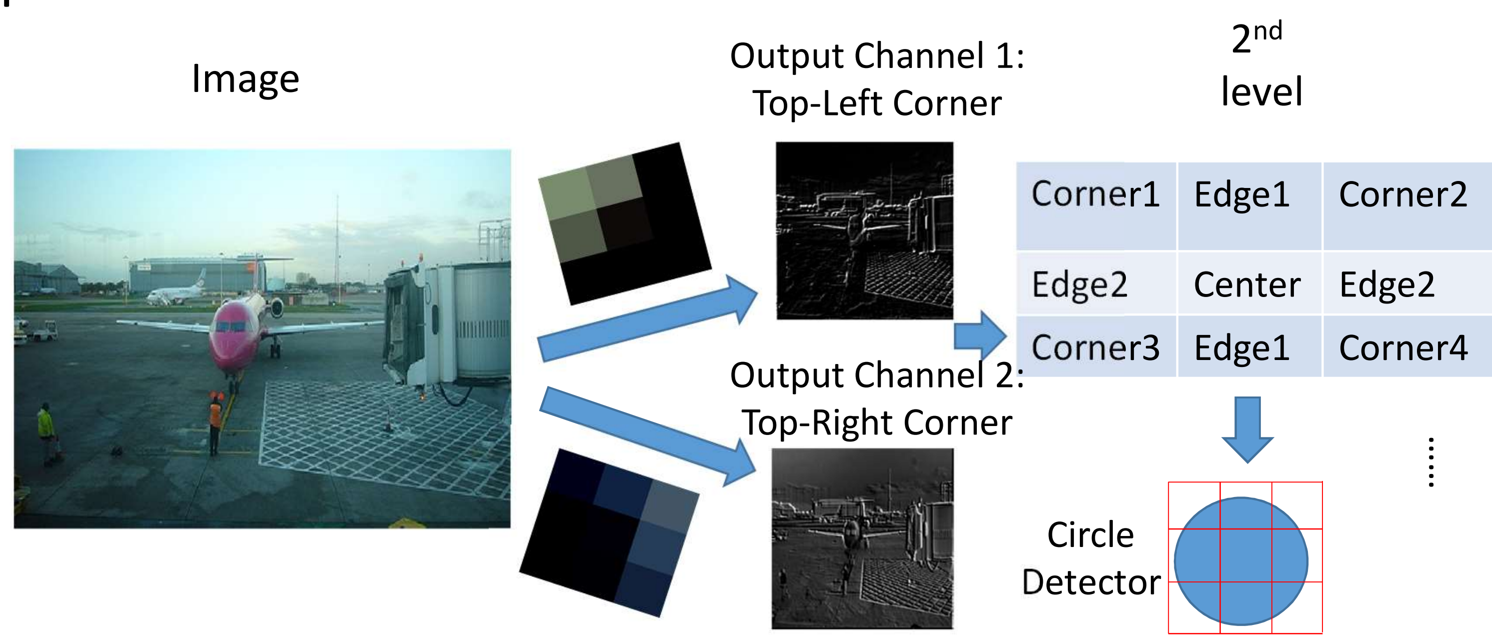
- In a color image:



- Each filter output goes to 1 channel

CNN: Multi-layer Architecture

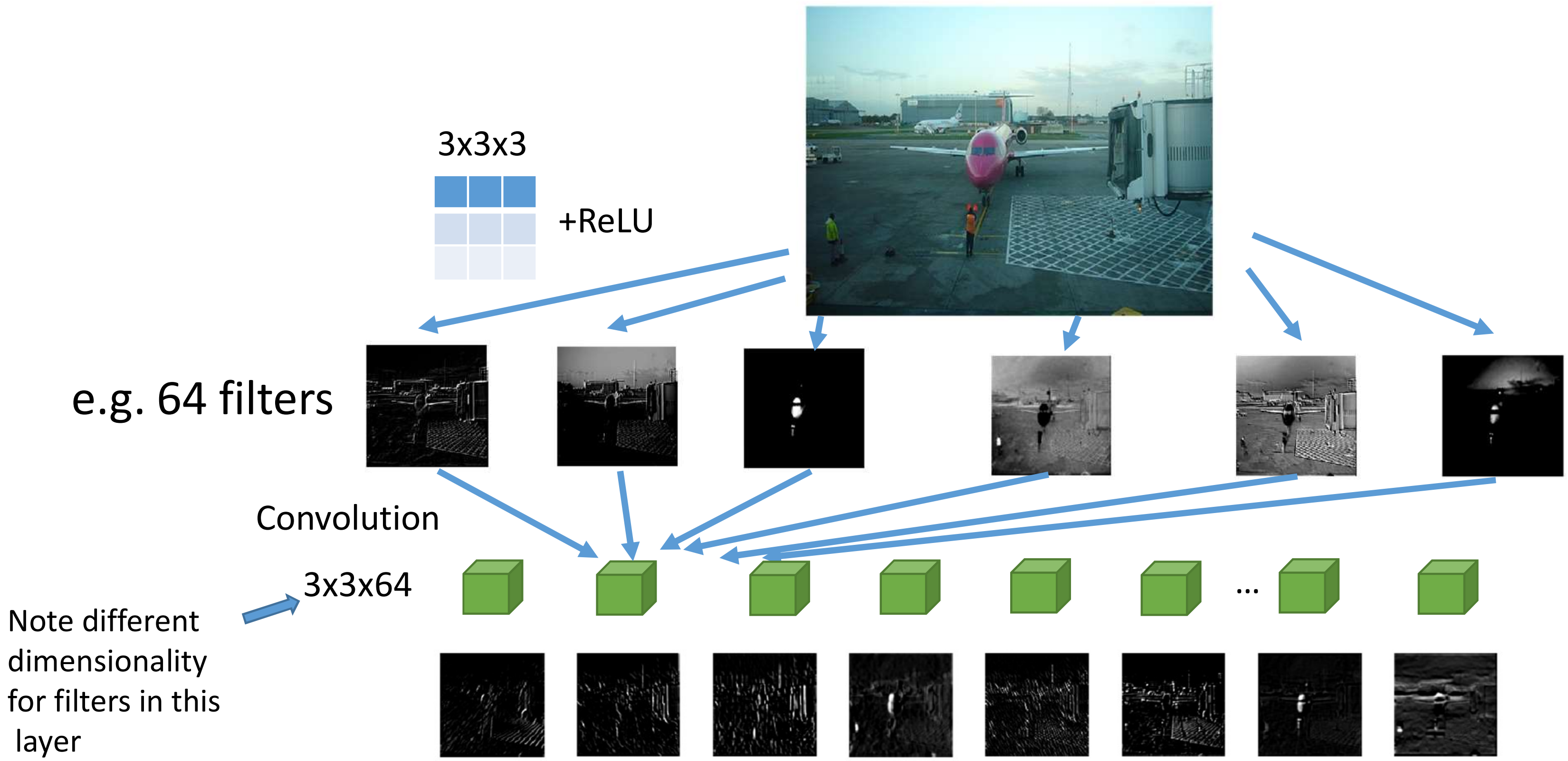
- Multi-layer architecture helps to generate more complicated templates



\newpage
\pagestyle{empty}

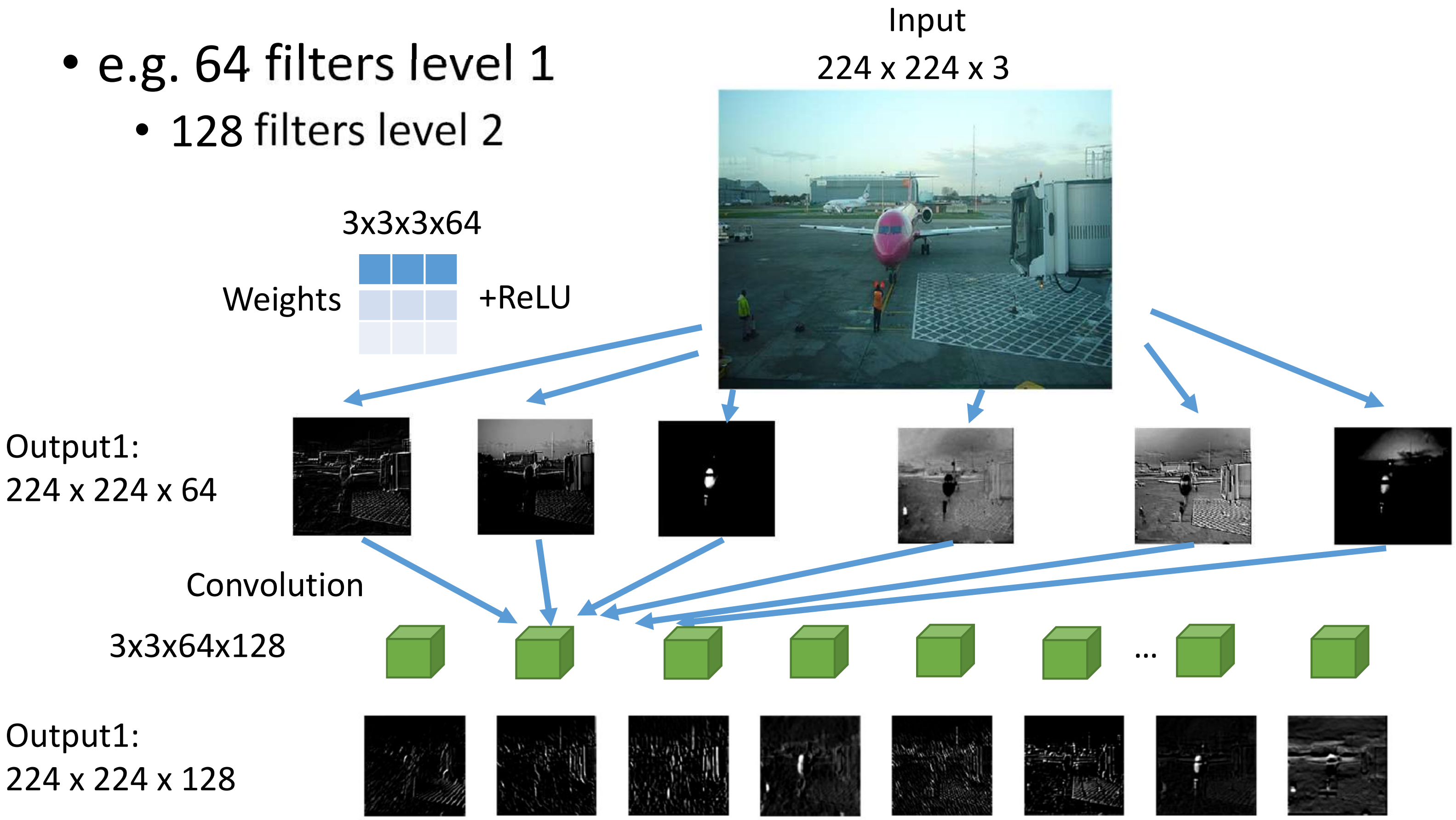
Convolutional Networks 2nd layer

- Each connection is a convolution



What's the shape of weights and input

- e.g. 64 filters level 1
  - 128 filters level 2



Dramatic reduction on the number of parameters

- Think about a fully-connected network on 256 x 256 image with 500 hidden units and 10 classes
  - Num. of params = 65536 * 3 * 500 + 500 * 10 = 98.3 Million
- 1-hidden layer convolutional network on 256 x 256 image with 11x11 and 500 hidden units?
  - Num. of params = 11 * 11 * 3 * 500 + 500 * 10 = 155,000
- 2-hidden layers convolutional network on 256 x 256 image with 11x11 – 3x3 sized filters and 500 hidden units in each layer?
  - Num. of params = 150,000 + 3 * 3 * 500 * 500 + 500 * 10 = 2.4 Million

Back to images

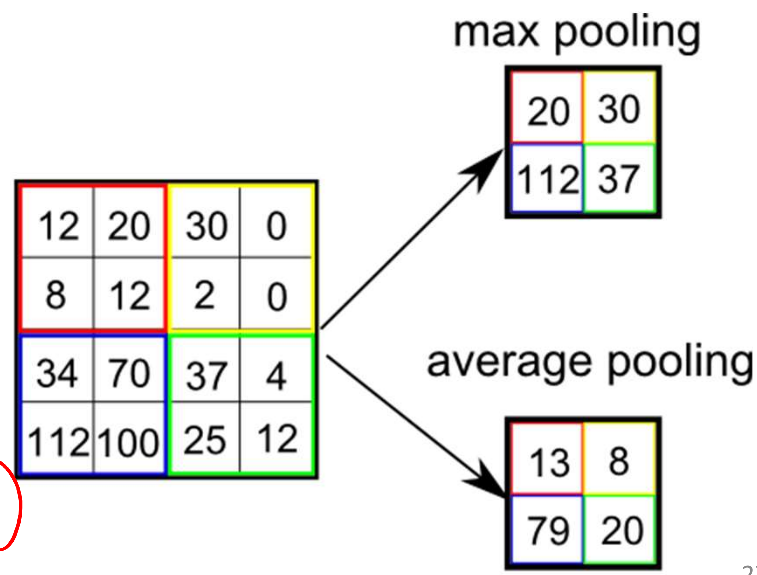
- Why images are much harder than digits?
- Much more deformation
- Much more noises
- Noisy background

Pooling

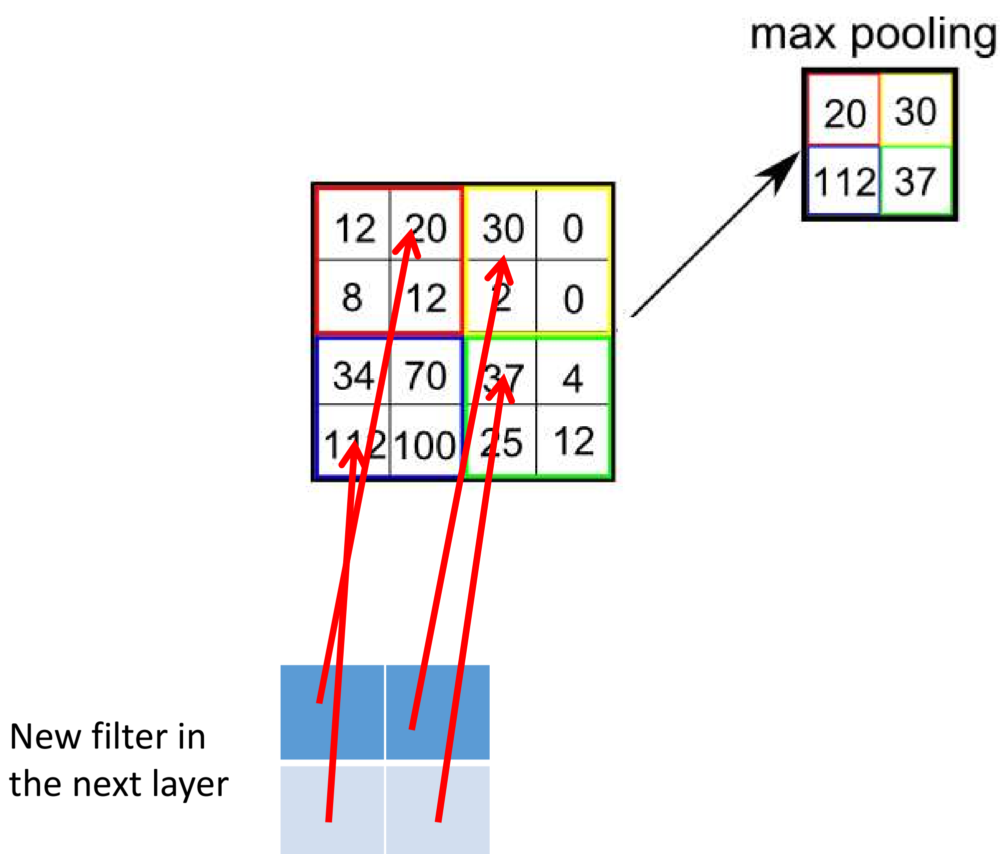
- Localized max-pooling (stride-2) helps achieving some location invariance
- As well as filtering out irrelevant background information

  e.g. $x_{out} = \max(x_{11}, x_{12}, x_{21}, x_{22})$

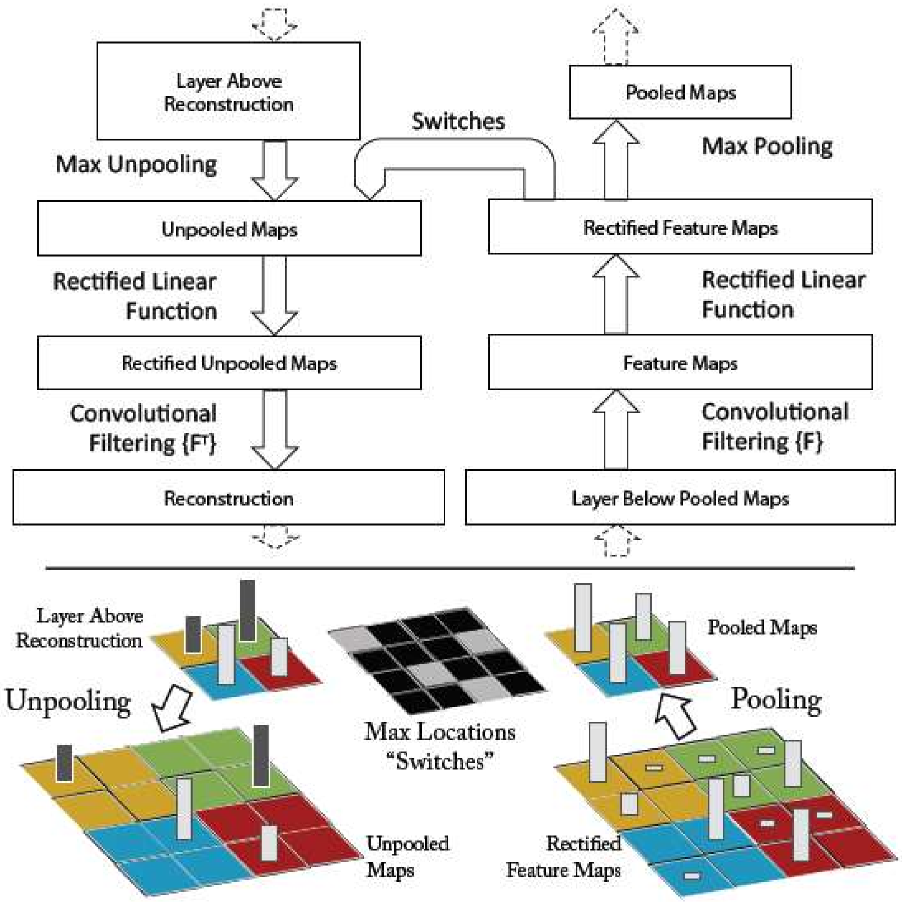
- What is the subgradient of this?



Deformation enabled by max-pooling



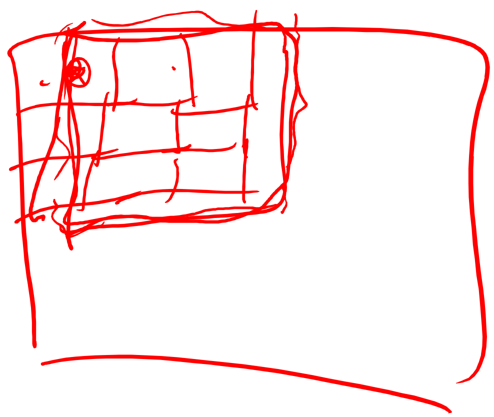
Deconvolutional Network

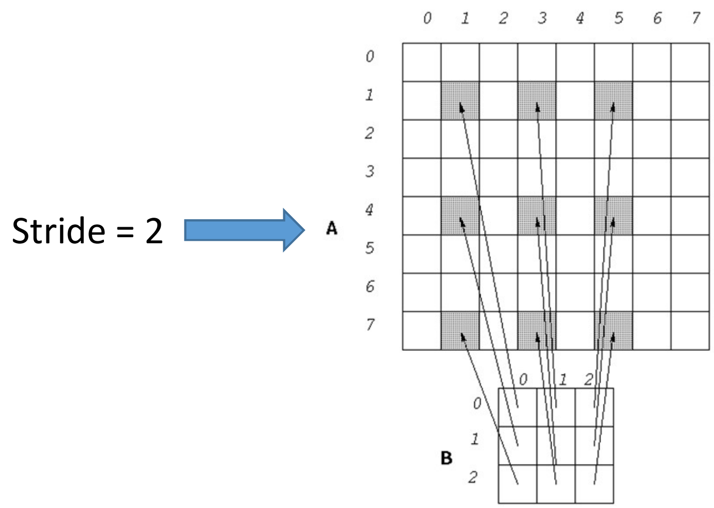


- **Instead of mapping pixels to features, map the other way around**
- **Reverts the max-pooling process**

Strides

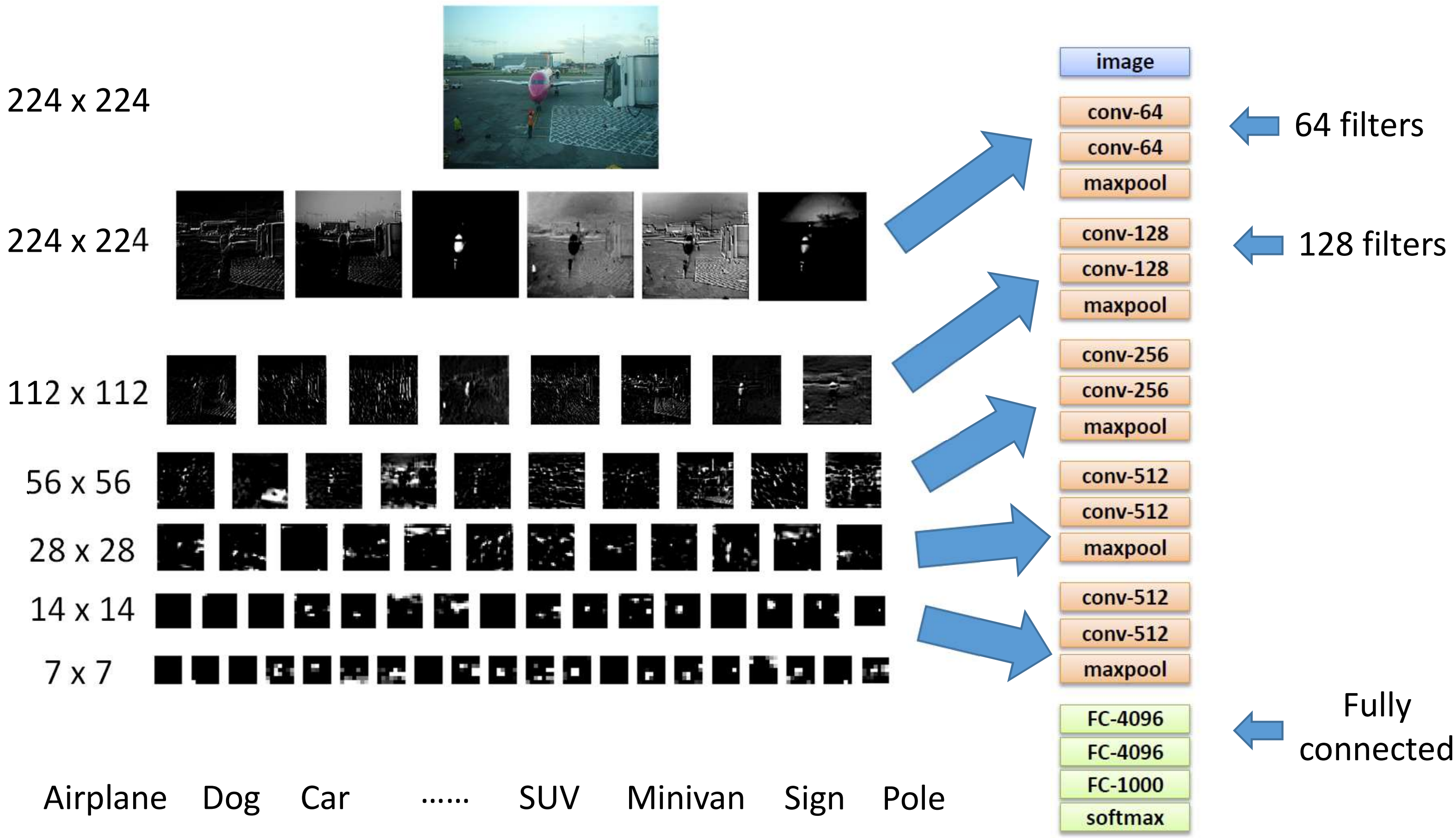
- Reduce image size by strides
  - Stride = 1, convolution on every pixel
  - Stride = 2, convolution on every 2 pixels
  - Stride = 0.5, convolution on every half pixel (interpolation, Long et al. 2015)





The VGG Network

$$\min_{W} \sum_{i=1}^{n} \text{softmax\_loss}(f_W(x_i), y_i)$$



(Simonyan and Zisserman 2014)

Why 224x224?

- The magic number 224 = 2^5 x 7, so that there is always a center-surround pattern in any layer

- Another potential candidate is 2^7 x 3 = 384
  - Some has shown larger is better
  - However more layers + bigger = more difficult to train, need more machines to tune parameters

Backpropagation for the convolution operator

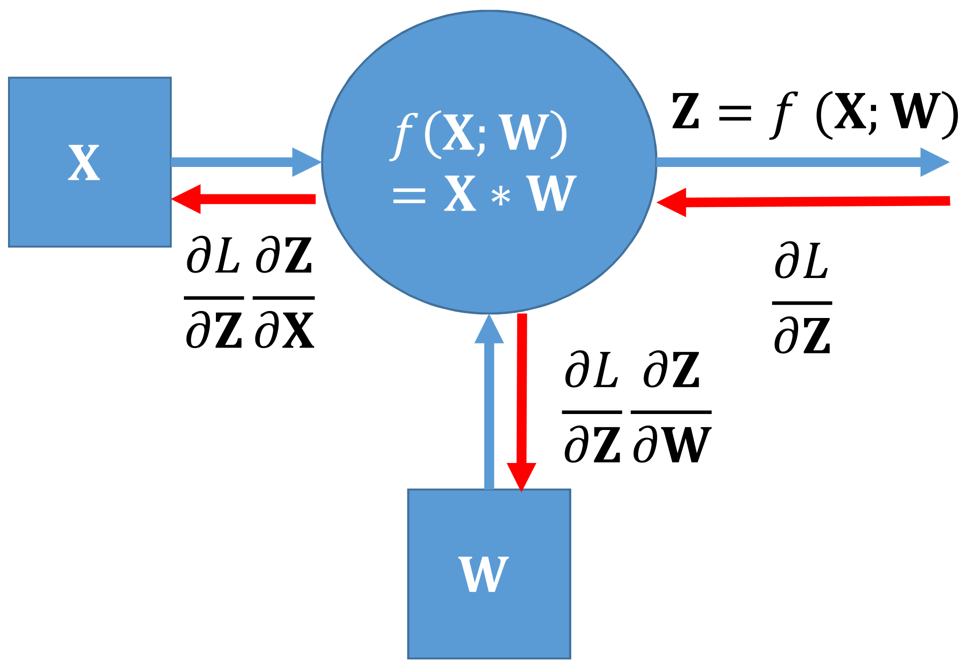
Forward pass:

Compute $f(X;W) = X * W$

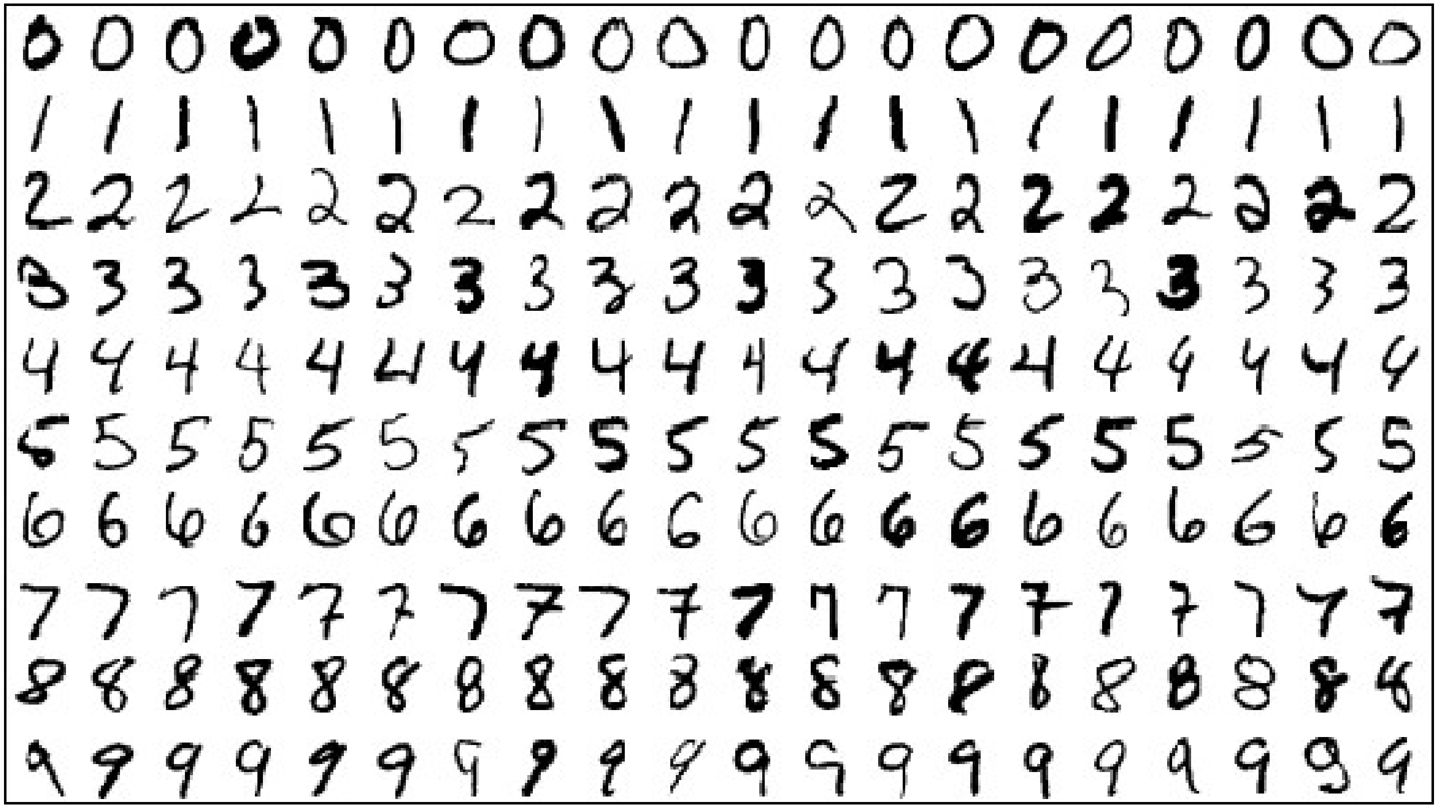
Backward pass:

Compute $\frac{\partial Z}{\partial X} = ?$

$\frac{\partial Z}{\partial W} = ?$



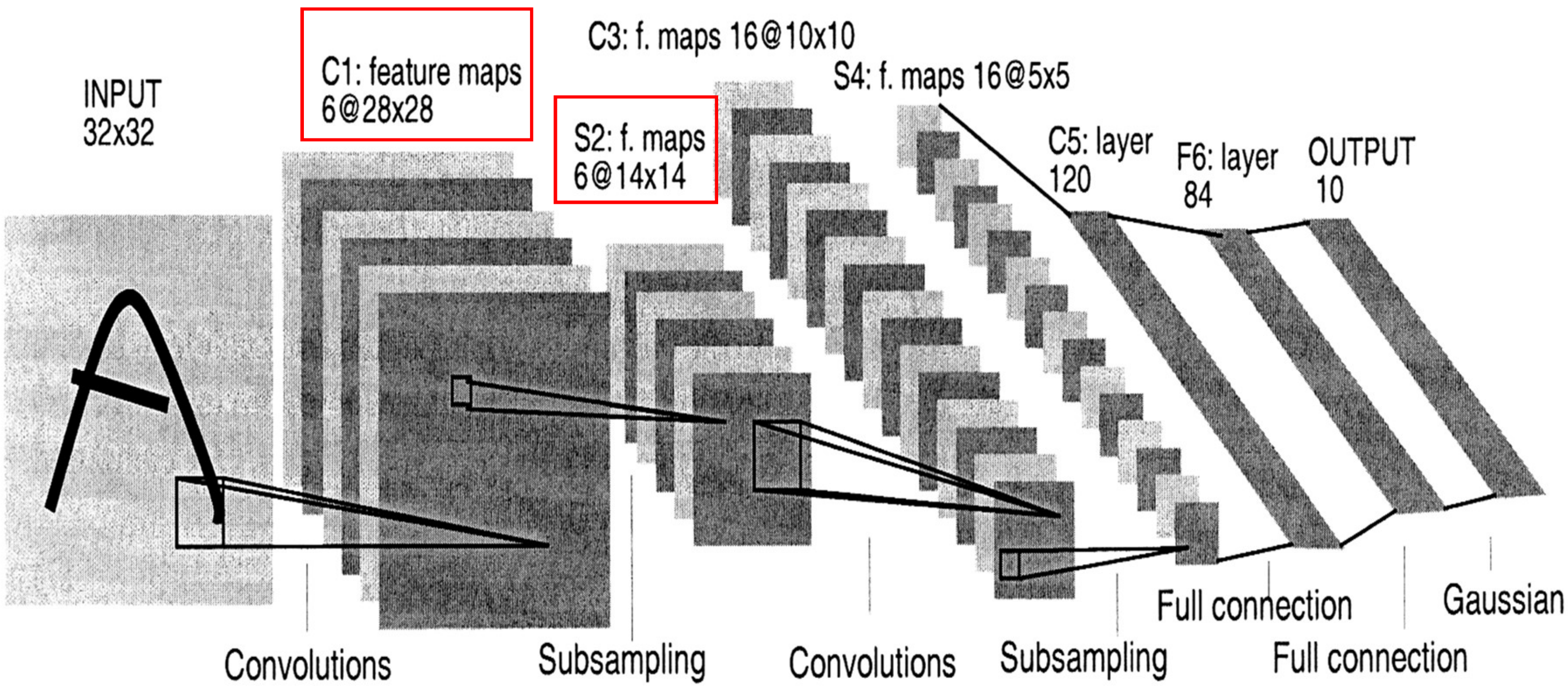
Historical Remarks: MNIST



Le Net

- Convolutional nets are invented by Yann LeCun et al. 1989
  - On handwritten digits classification
  - Many hidden layers
  - Many maps of replicated units in each layer.
  - Pooling of the outputs of nearby replicated units.
  - A wide net that can cope with several characters at once even if they overlap.
  - A clever way of training a complete system, not just a recognizer.
- This net was used for reading ~10% of the checks in North America.
- Look the impressive demos of LENET at http://yann.lecun.com

The architecture of LeNet5 (LeCun 1998)



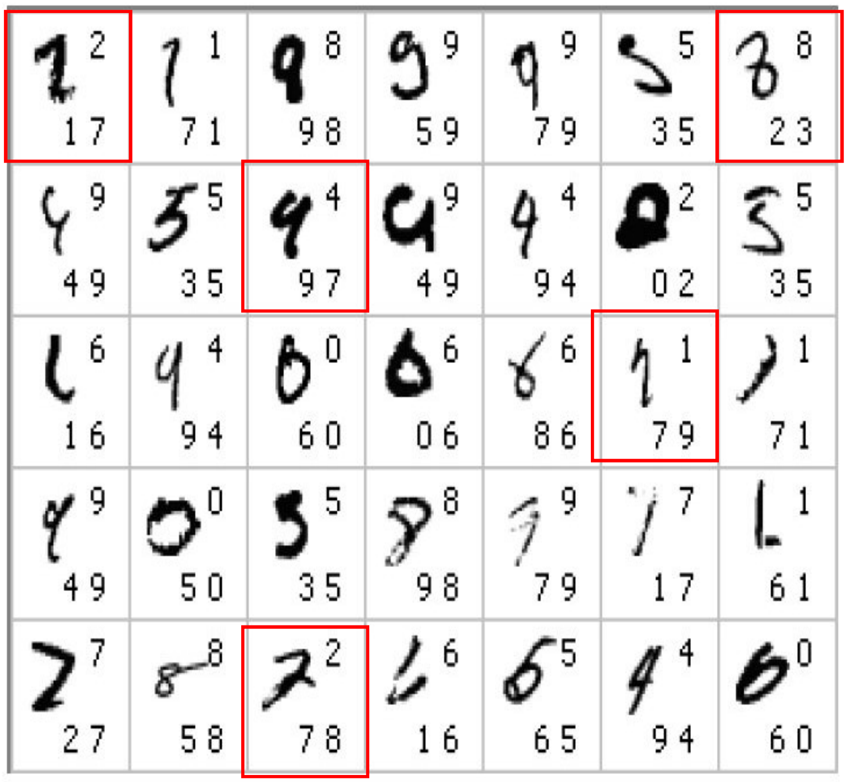
ConvNets performance on MNIST

Convolutional net LeNet-1	subsampling to 16x16 pixels	1.7	LeCun et al. 1998
Convolutional net LeNet-4	none	1.1	LeCun et al. 1998
Convolutional net LeNet-4 with K-NN instead of last layer	none	1.1	LeCun et al. 1998
Convolutional net LeNet-4 with local learning instead of last layer	none	1.1	LeCun et al. 1998
Convolutional net LeNet-5, [no distortions]	none	0.95	LeCun et al. 1998
Convolutional net, cross-entropy [elastic distortions]	none	0.4	Simard et al., ICDAR 2003

The 82 errors made by LeNet5

The human error rate is probably about 0.2% - 0.3% (quite clean)

The errors made by the Ciresan *et. al.* net



The top printed digit is the right answer. The bottom two printed digits are the network's best two guesses.

The right answer is almost always in the top 2 guesses.

With model averaging they can now get about 25 errors.

What's different from back then till now

- Computers are bigger, faster
- GPUs

	LeNet (1989)	LeNet (1998)	AlexNet (2012)
classification task	digits	digits	objects
categories	10	10	1,000
image size	16×16	28×28	$256 \times 256 \times 3$
training examples	7,291	60,000	1.2 million
units	1,256	8,084	658,000
parameters	9,760	60,000	60 million
connections	65,000	344,000	652 million
total operations	11 billion	412 billion	200 quadrillion (est.)

What else is different?

- ReLU rectifier
- Max-pooling
  - Grab local features and make them global
- Dropout regularization (to-be-discussed)
  - Replaceable by some other regularization techniques

ReLU vs. Sigmoid