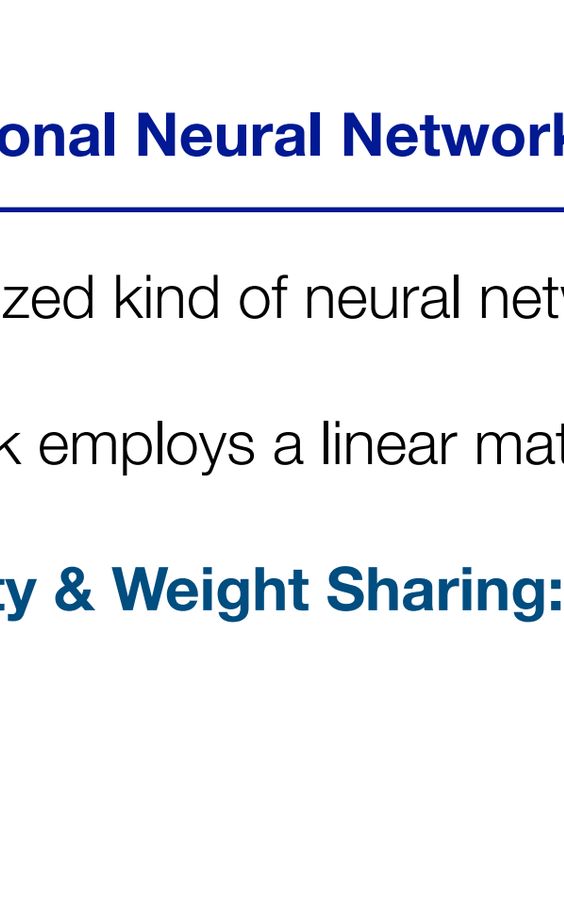
Deep Learning with Neural Networks

Introduction to Convolutional Neural Networks

Original: Pablo Martínez Olmos, pamartin@ing.uc3m.es
Adapted: Alejandro Guerrero López, alguerre@pa.uc3m.es

Convolutional Neural Networks

- Specialized kind of neural network for processing **data that has a known grid-like** topology
- Network employs a linear mathematical operation called **convolution**
- **Sparsity & Weight Sharing:** efficient front-end for NN feedforward structures



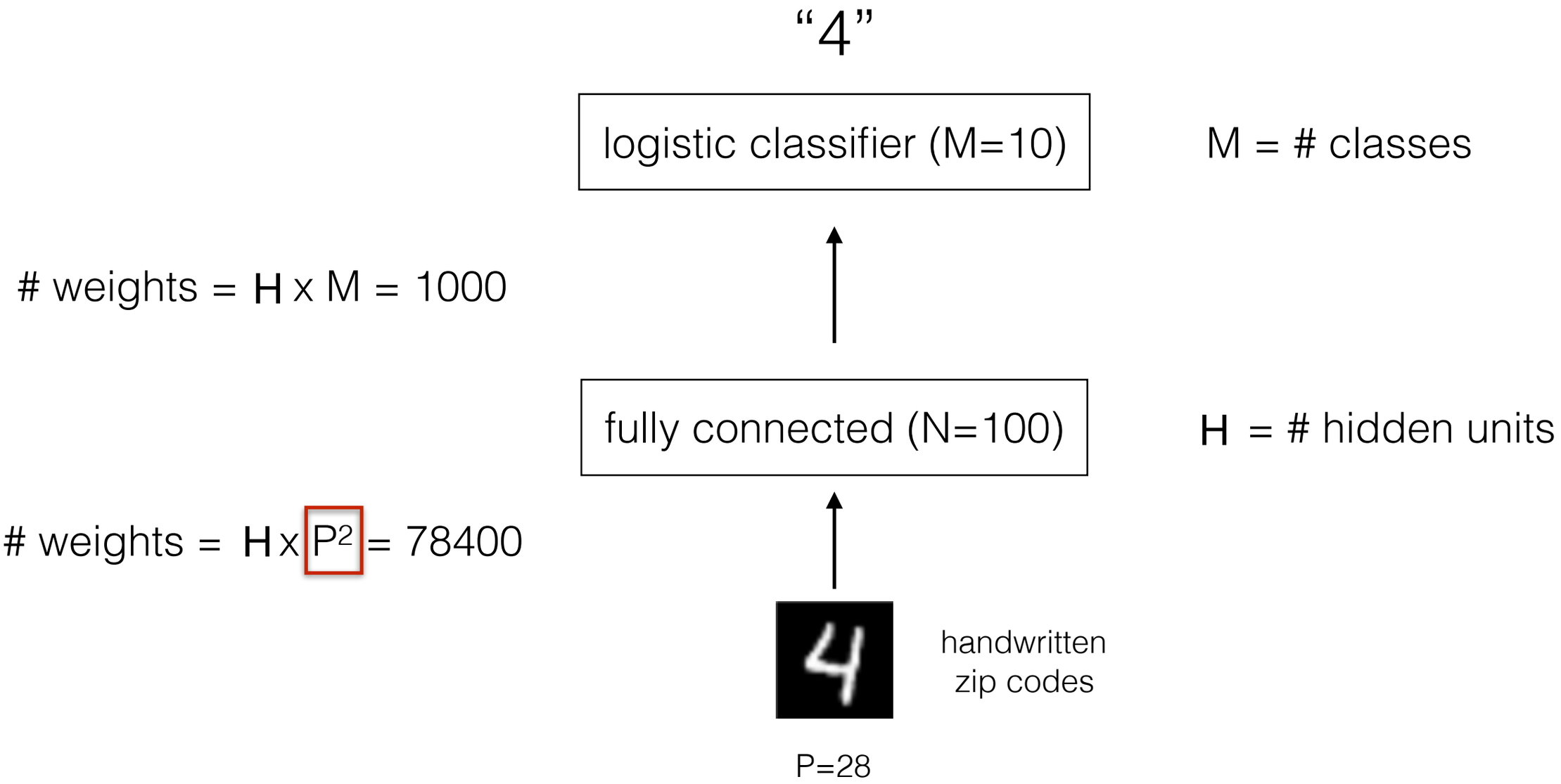
What the computer sees

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

An image is just a matrix of numbers [0,255]!
i.e., 1080x1080x3 for an RGB image

MLPs



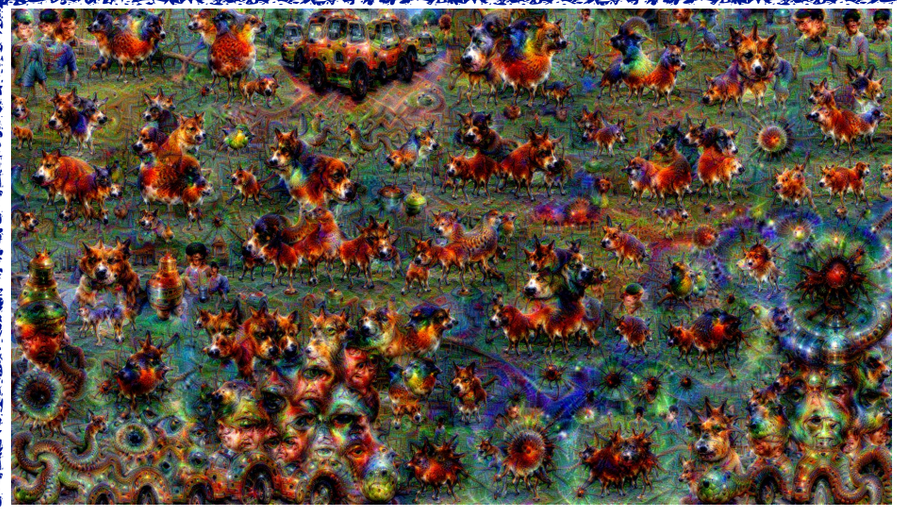
Note that weights grow as the **square of the number of pixels**

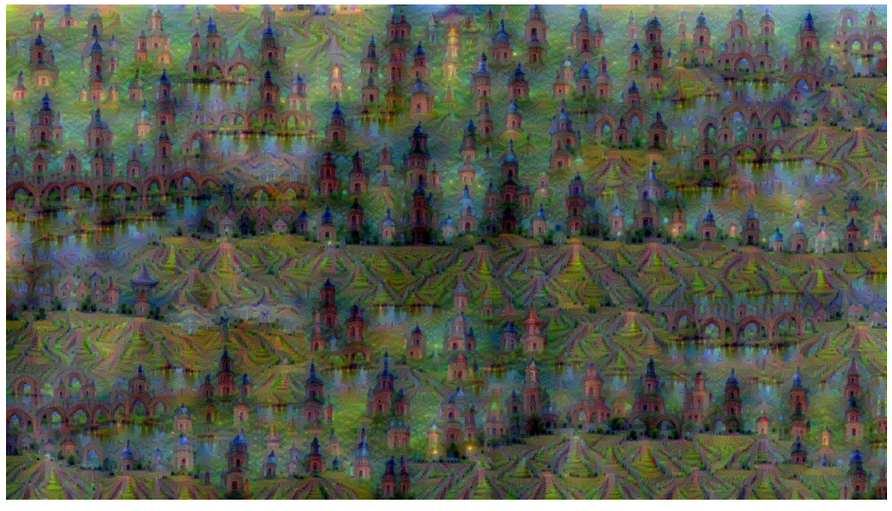
Unfeasible for large images!

Convolutional Neural Networks

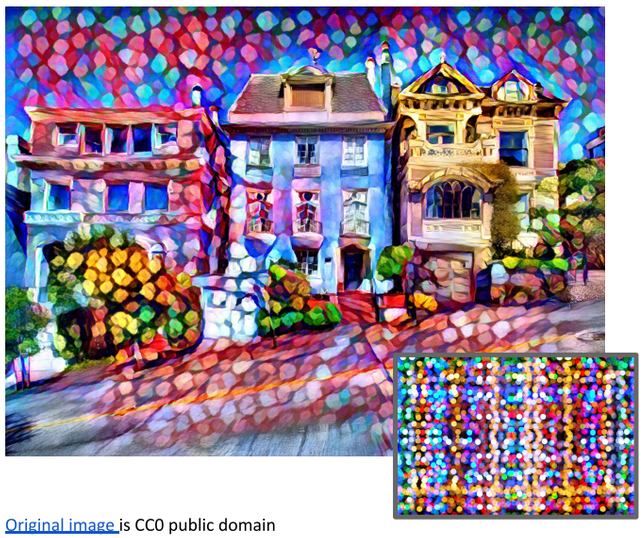
Classification

Retrieval





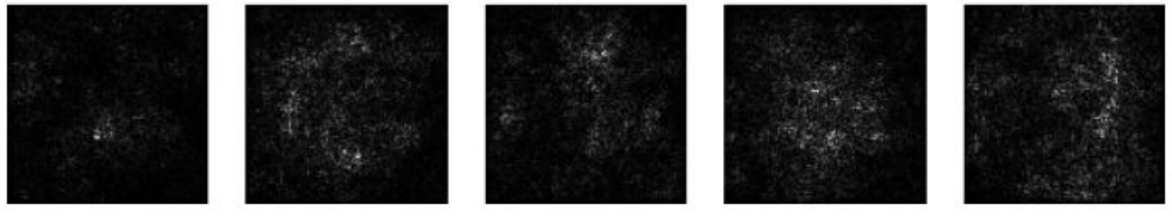
Figures copyright Justin Johnson, 2015. Reproduced with permission. Generated using the Inceptionism approach from a blog post by Google Research.



Original image is CC0 public domain
Starry Night and Tree Roots by Van Gogh are in the public domain
Bokeh image is in the public domain
Stylized images copyright Justin Johnson, 2017;

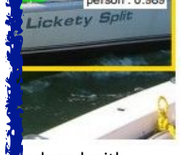
Gatys et al, "Image Style Transfer using Convolutional Neural Networks", CVPR 2016
Gatys et al, "Controlling Perceptual Factors in Neural Style Transfer", CVPR 2017

Benign



[Levy et al. 2016]

Figure copyright Levy et al. 2016.
Reproduced with permission.



roduced with

2015]

Figures copyright Clement Farabet, 2012.
Reproduced with permission.

[Farabet et al., 2012]

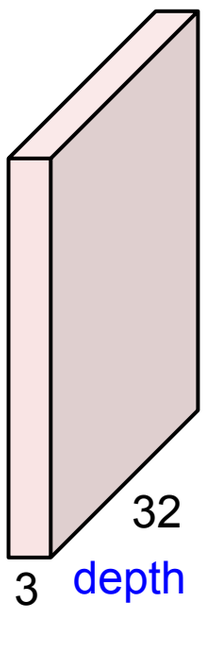
Natural image statistics obey invariances

ConvNet architectures make the explicit assumption that the inputs are images, which allows us to **encode certain properties into the architecture**.

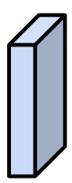
…
translation
cropping
dilation
contrast
rotation
scale
brightness
…

Convolution layers

32x32x3 image -> preserve spatial structure



5x5x3 filter



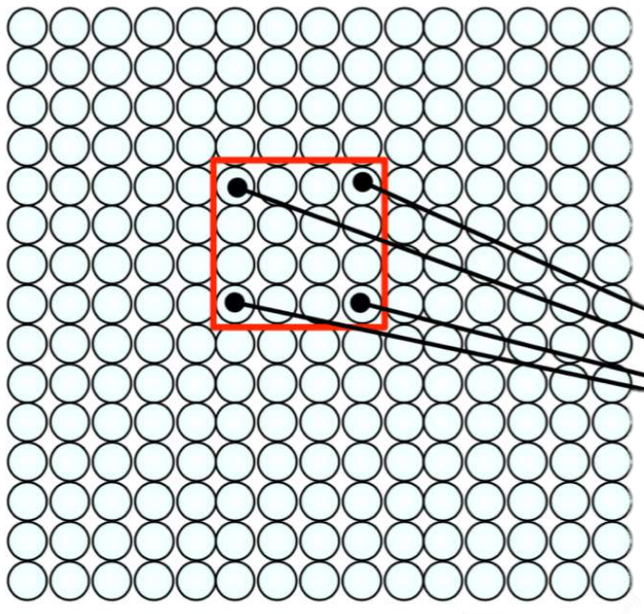
Convolve the filter with the image
i.e. “slide over the image spatially,
computing dot products”

CNN animations

@Stanford CS231n: Convolutional Neural Networks for Visual Recognition

Using Spatial Structure

Input: 2D image.
Array of pixel values



Idea: connect patches of input
to neurons in hidden layer.

Neuron connected to region of
input. Only “sees” these values.

Convolution layers

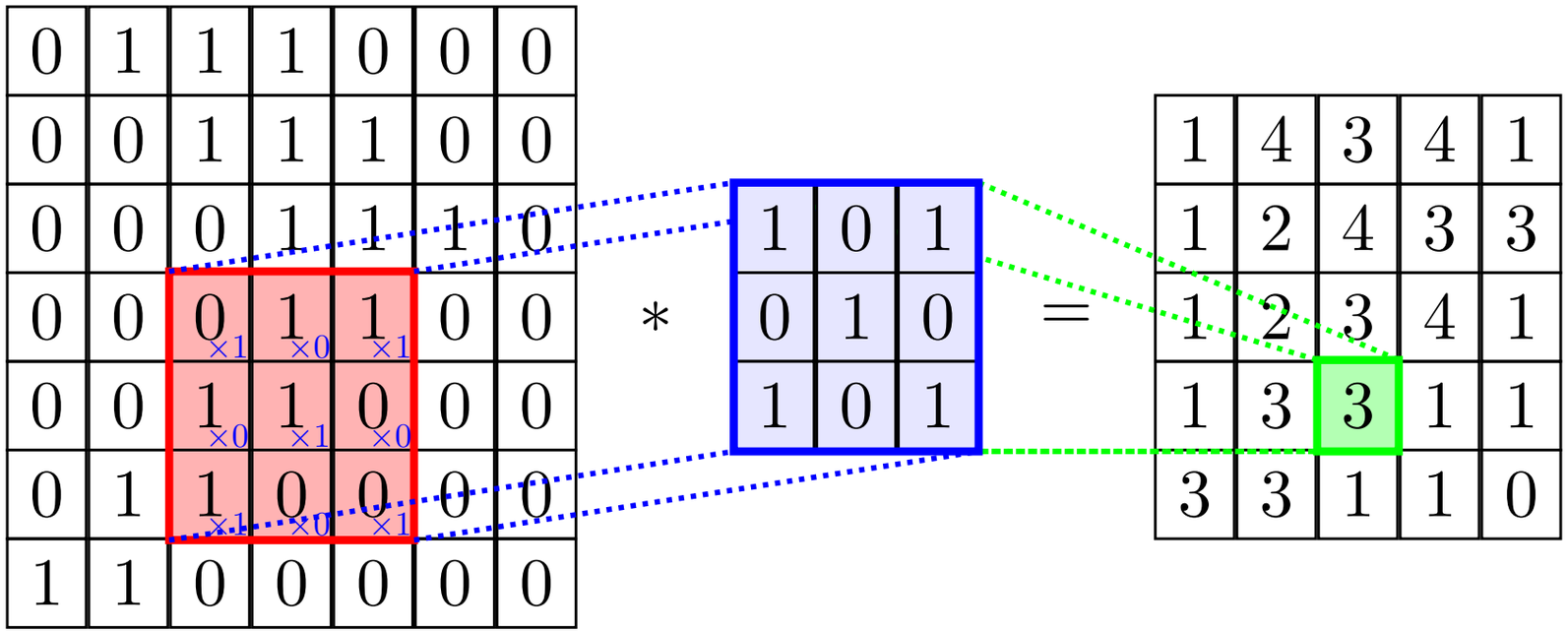
original

filter (3 x 3)

	0	1	0	
	1	-4	1	
	0	1	0	

all edge detector

Convolution layers



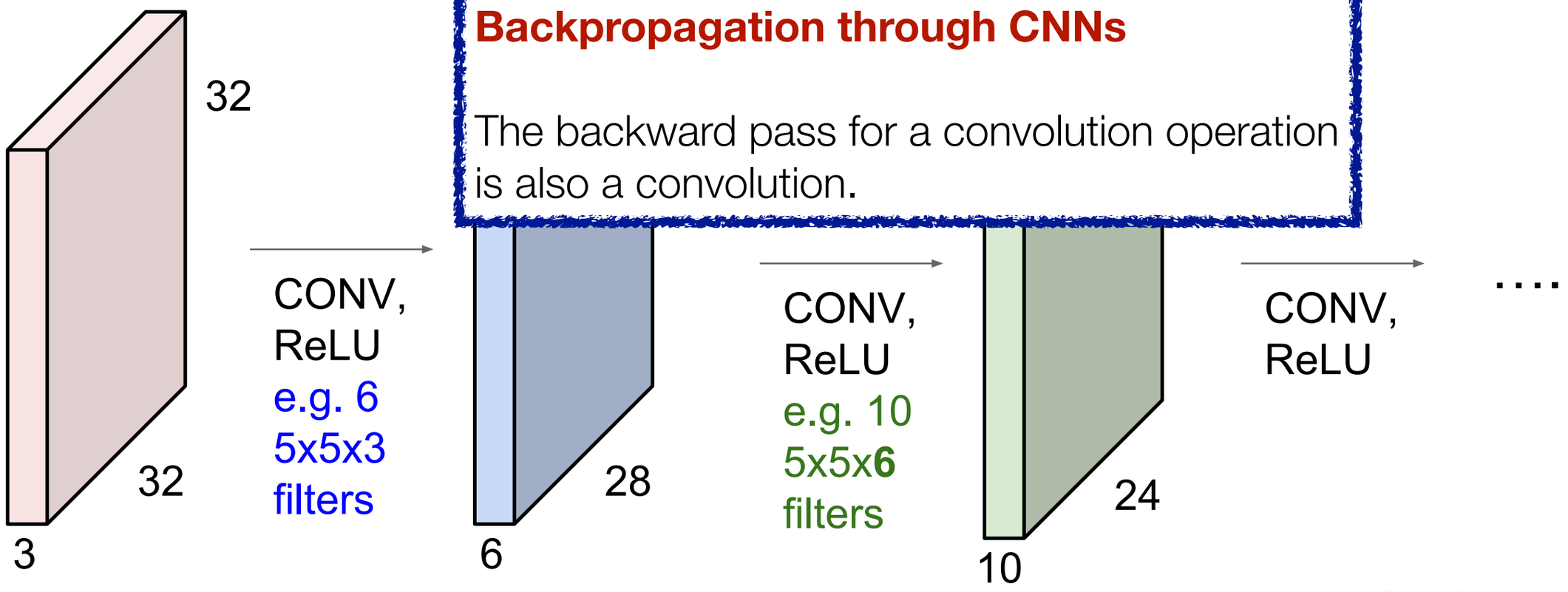
$X_{n\times n}$ K Y

$$Y[i,j] = \sum_{k_1=1}^{n}\sum_{k_2=1}^{n} X[k_1,k_2]K[i-k_1,j-k_2]$$

$$K[u,q] = 0 \qquad u > 3, q > 3$$

ConvNets

Use **multiple filters** to extract different features. **Spatially share** parameters of each filter (features that matter in one part of the input should matter elsewhere)



Backpropagation through CNNs

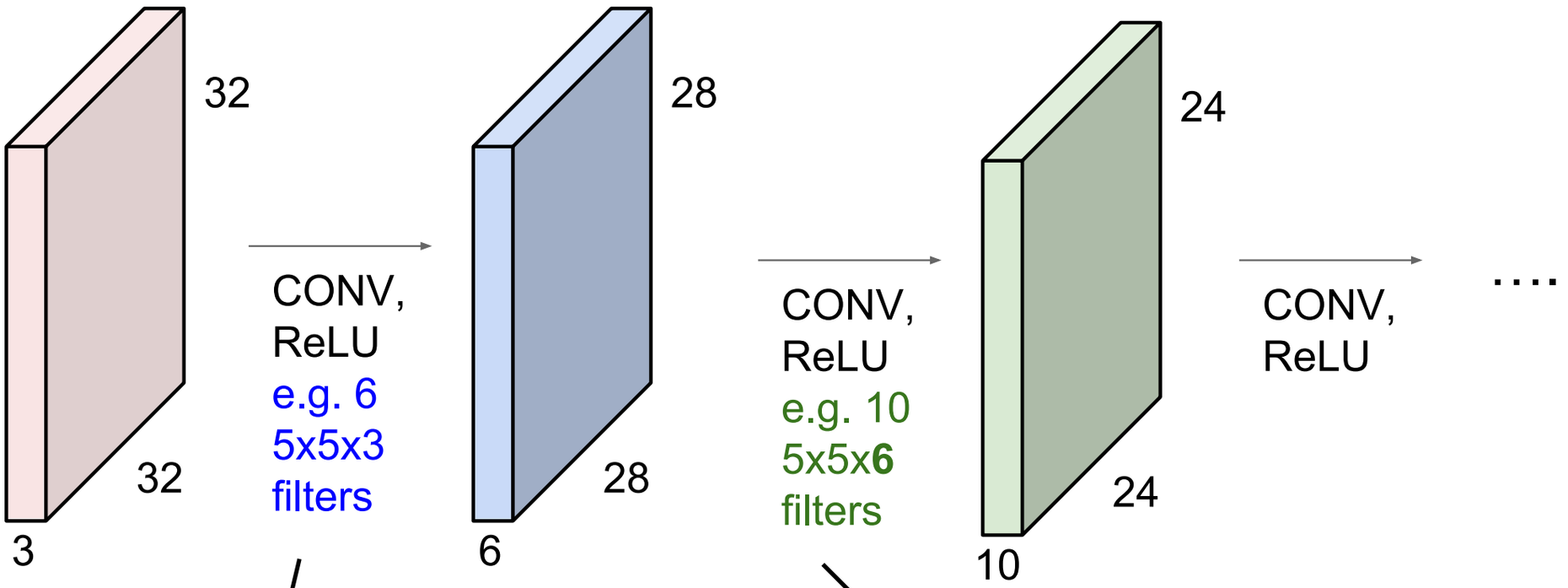
The backward pass for a convolution operation is also a convolution.

We stack these up to get a "new image" of size 28x28x6

ConvNet is a **sequence of Convolution Layers**, interspersed with **activation functions**

ConvNets reduce spatial dimension as you go deeper

Stride=1



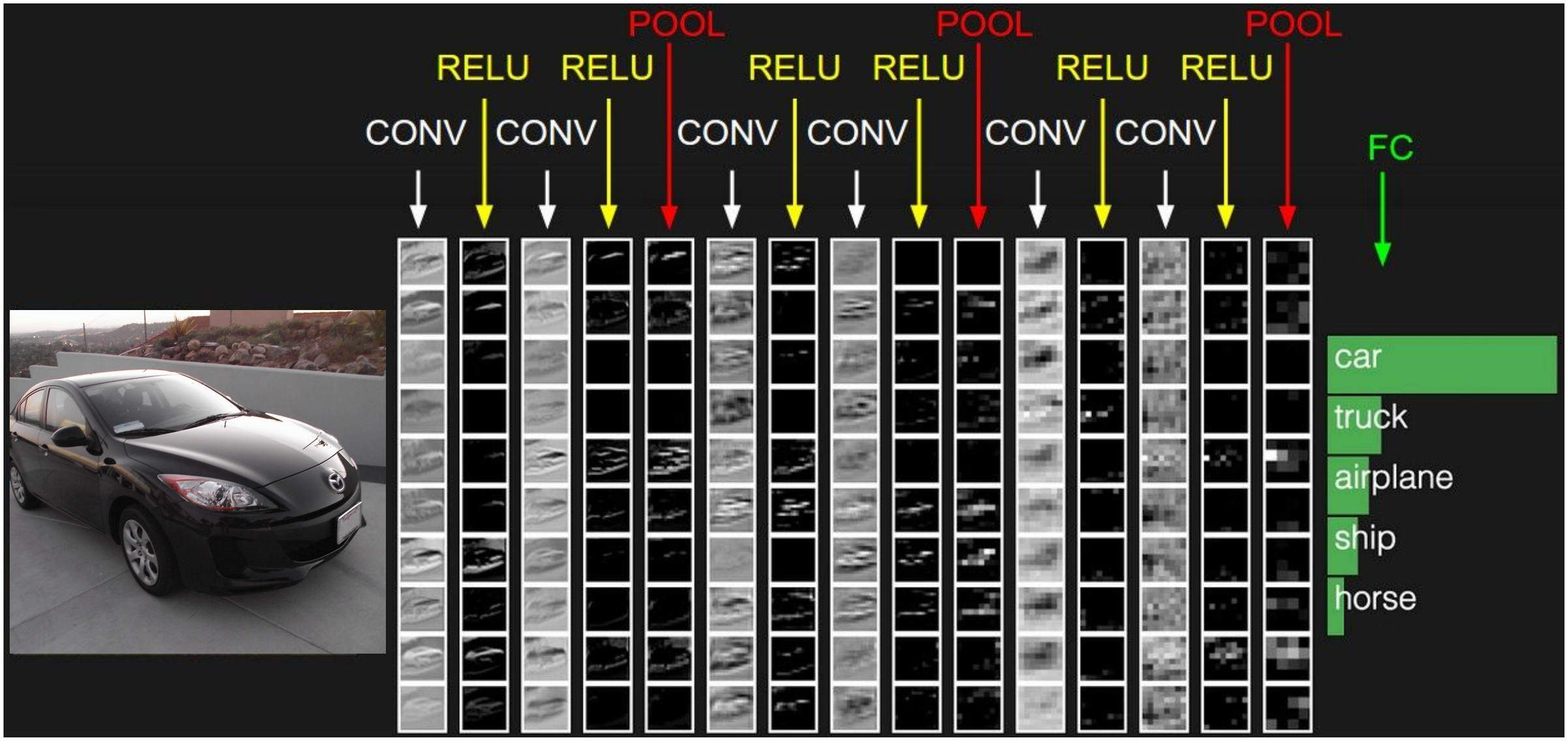
5x5x3+3=78 parameters per filter
(+ 3 for the bias vector)

6 x 78 = 468 parameters

5x5x6+6=156 parameters per filter

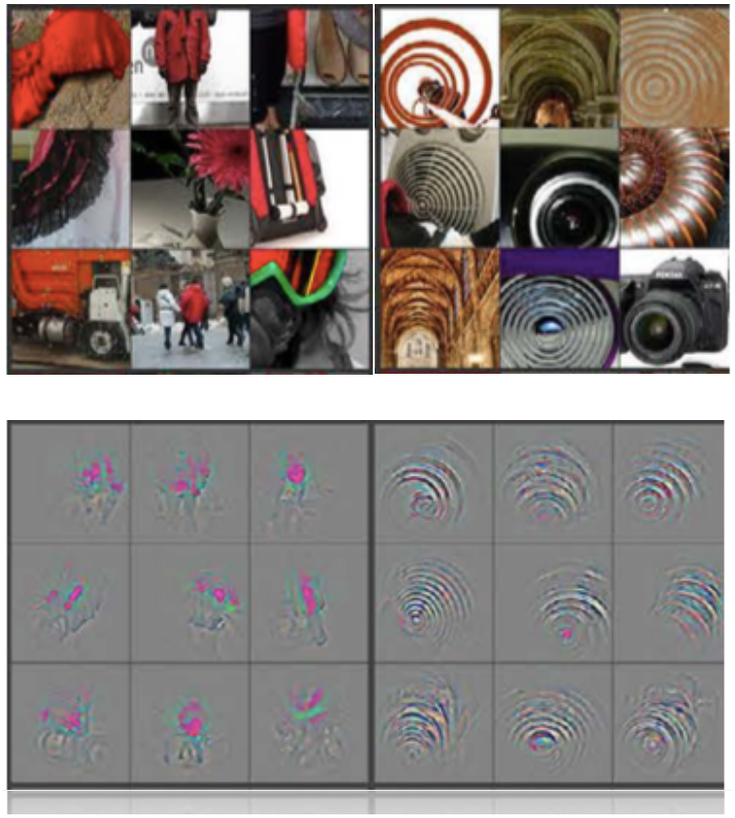
10 x 156 = 1560 parameters

Representation Learning in ConvNets

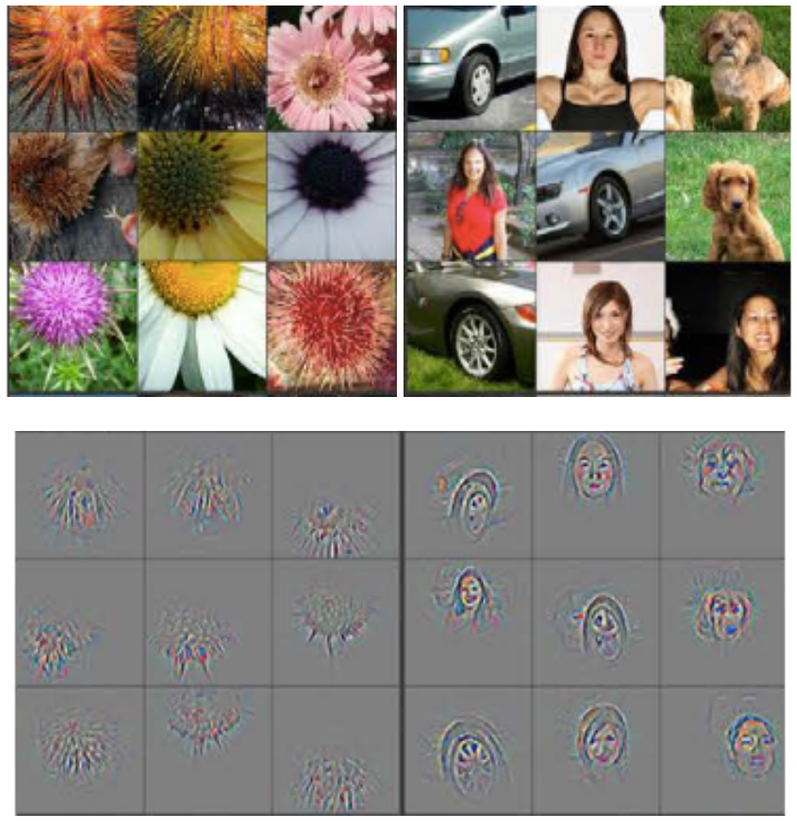


Representation Learning in ConvNets

layer 3

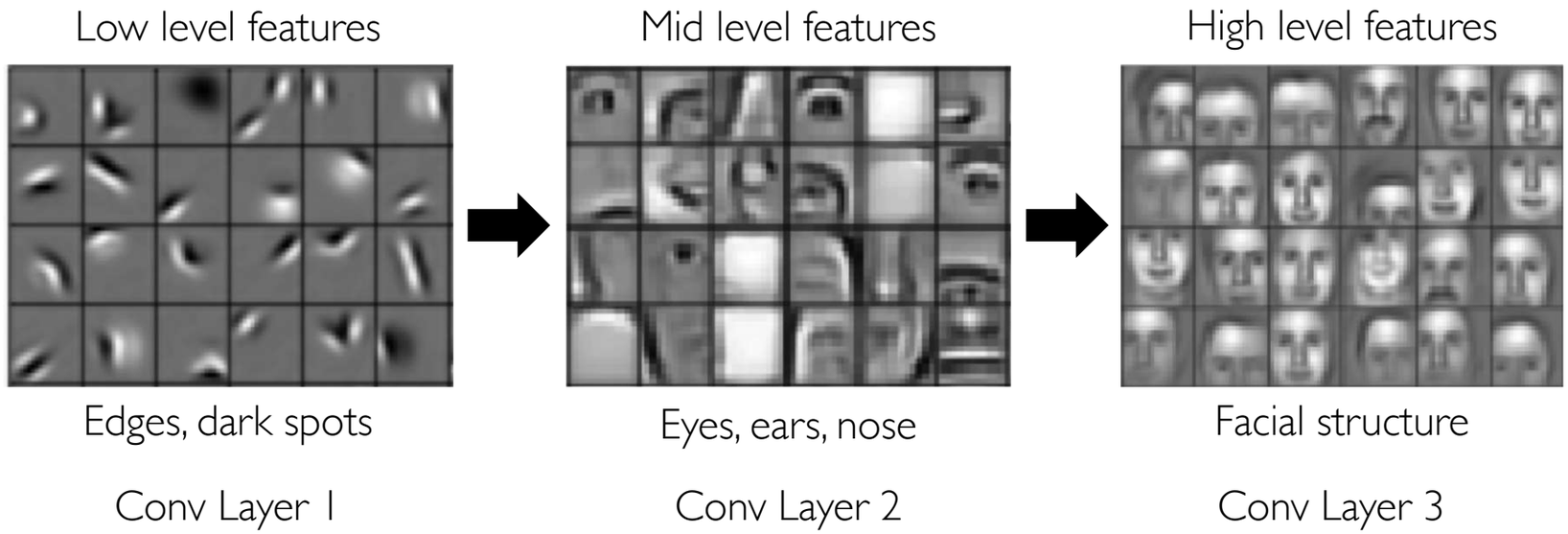


layer 5



Find image pixels that elicit largest activation

Visualizing and Understanding Convolutional Networks
M Zeiler and R Fergus (2013)



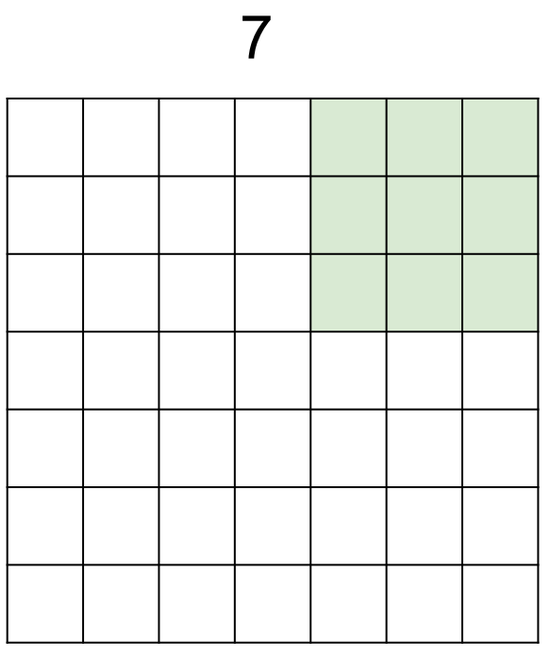
A closer look to spatial dimensions. Stride & Zero padding

7x7 input

Stride: Filter Step Size

Stride = 1

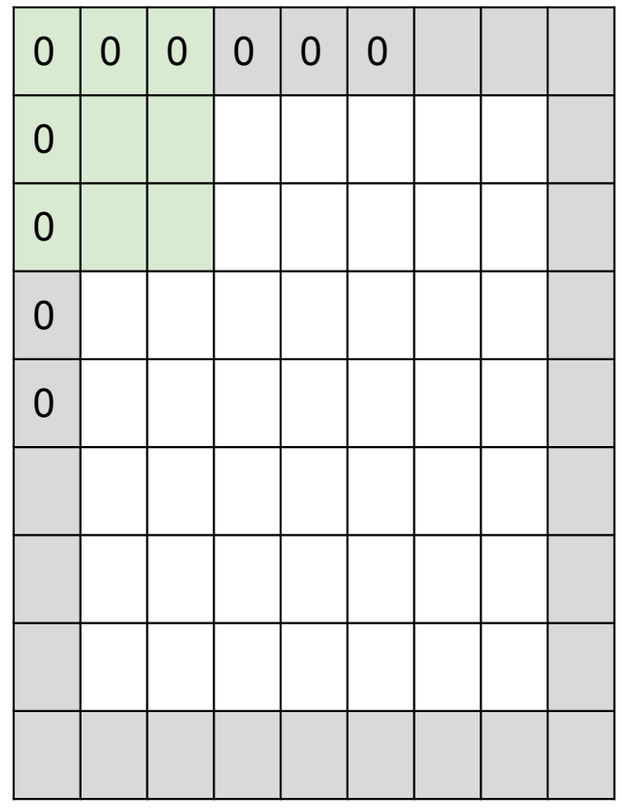
5x5 Output



Stride = 2

3x3 Output

In practice: zero-pad the image when needed

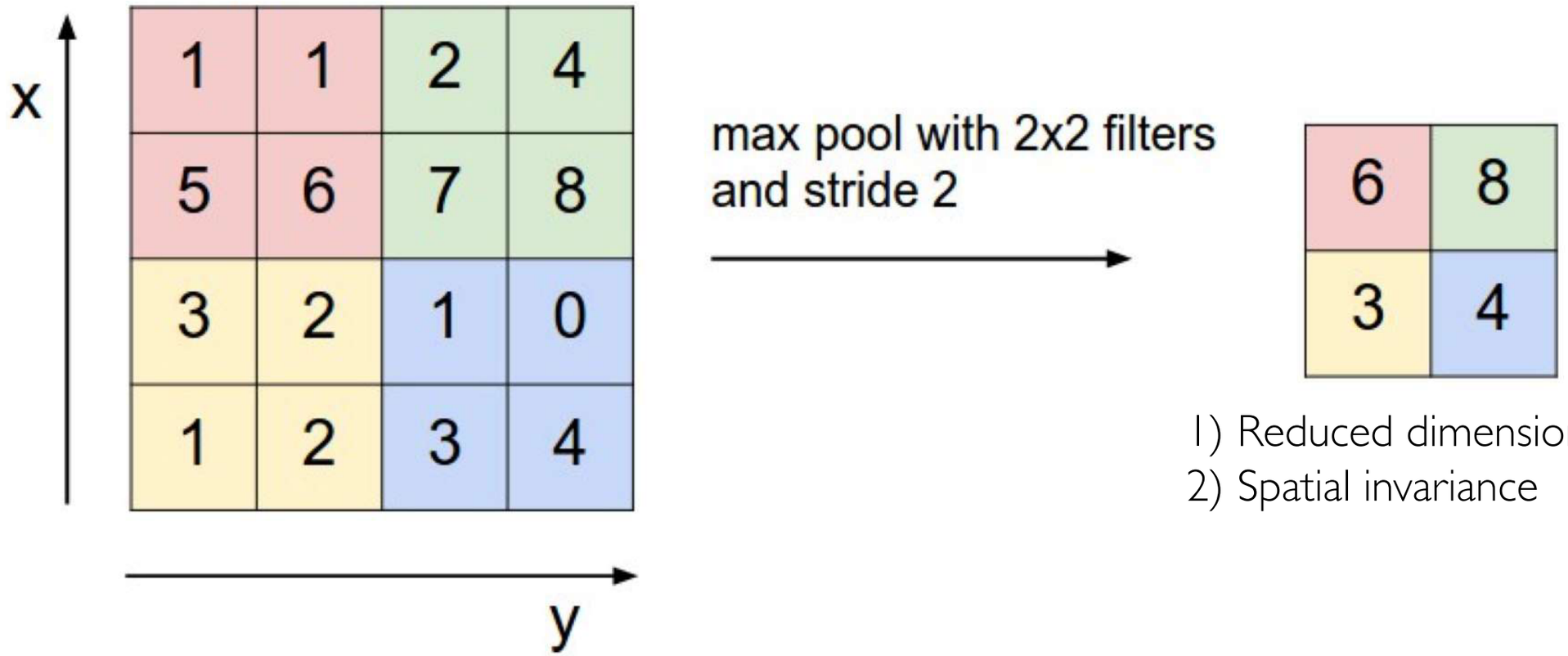


Output size:

$$(N - F)/\text{stride} + 1$$

Pooling Layers

Alternatively to stride >1, we can use *pooling layers*

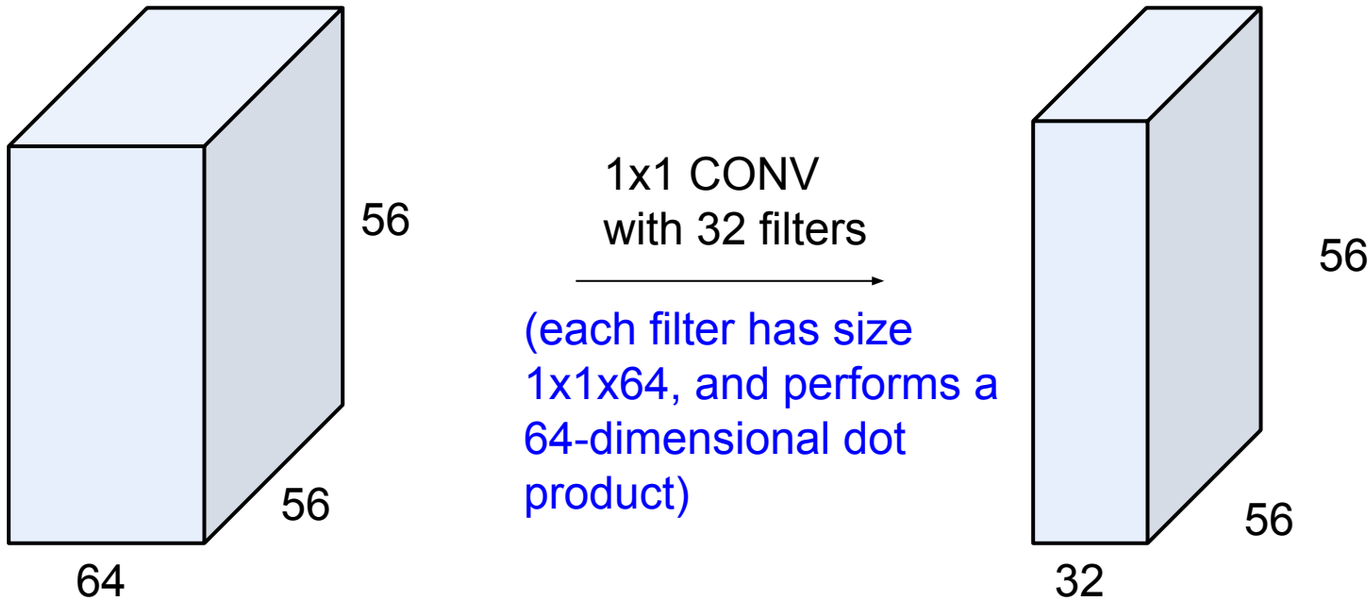


1) Reduced dimensionality
2) Spatial invariance

Biggest reason use of pooling is historical.
Common pretrained models use pooling, so finetuned and derived models use pooling too.

1x1 convolutions

Stride=1



1x1 convolution leads to dimension reductionality

Case Studies

LeNet

PROC. OF THE IEEE, NOVEMBER 1998

Gradient-Based Learning Applied to Document Recognition

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner

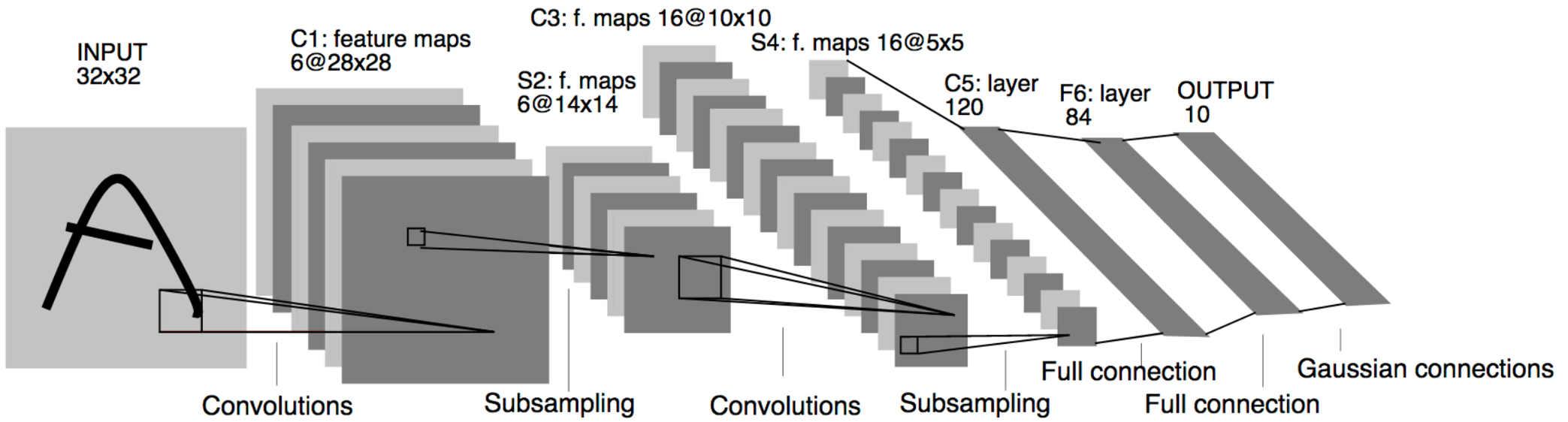


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Used to read zip codes, digits, …

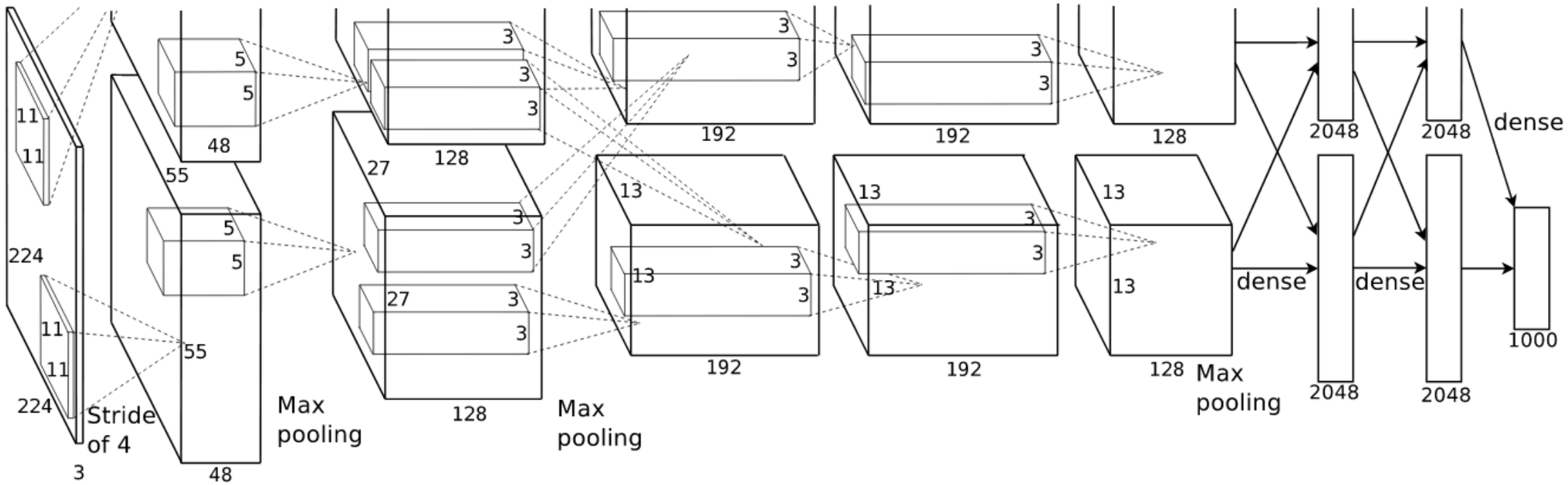
AlexNet

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



Submitted to the **ImageNet ILSVRC challenge** in 2012 and significantly outperformed the second runner-up (top 5 error of 16% compared to runner-up with 26% error).

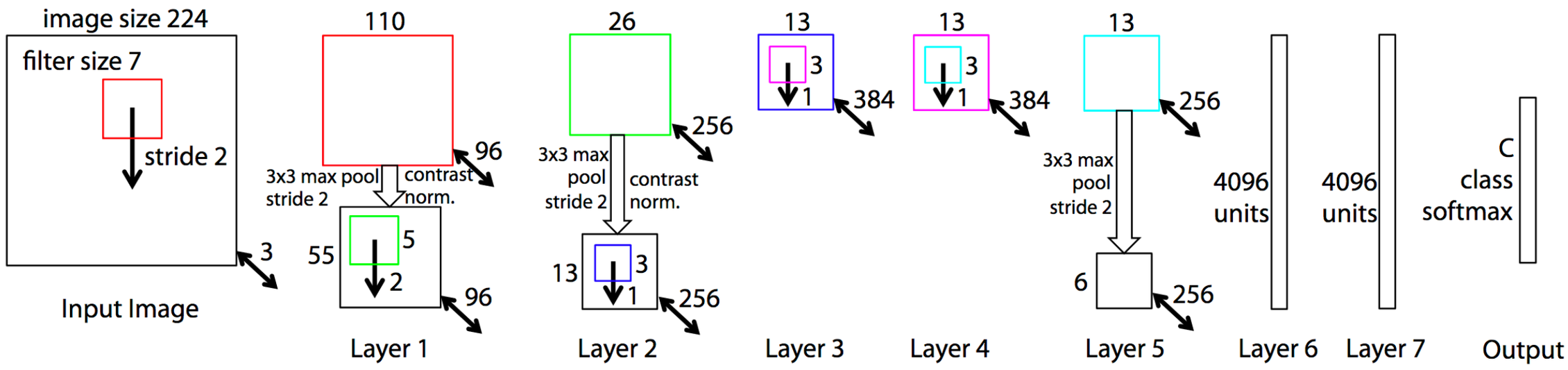
Similar architecture to LeNet, but was **deeper, bigger, and featured Convolutional Layers stacked on top of each other**

ZF Net

Visualizing and Understanding Convolutional Networks

Matthew D. Zeiler ZEILER@CS.NYU.EDU
Dept. of Computer Science, Courant Institute, New York University

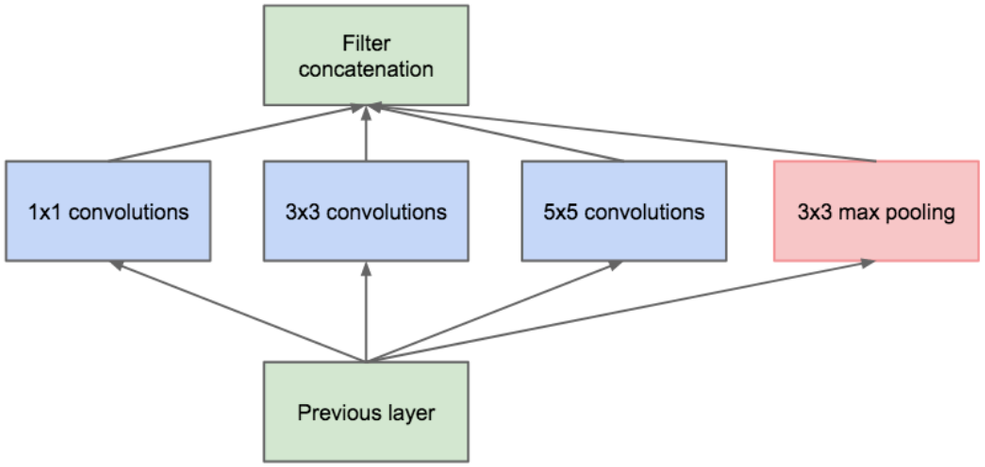
Rob Fergus FERGUS@CS.NYU.EDU
Dept. of Computer Science, Courant Institute, New York University



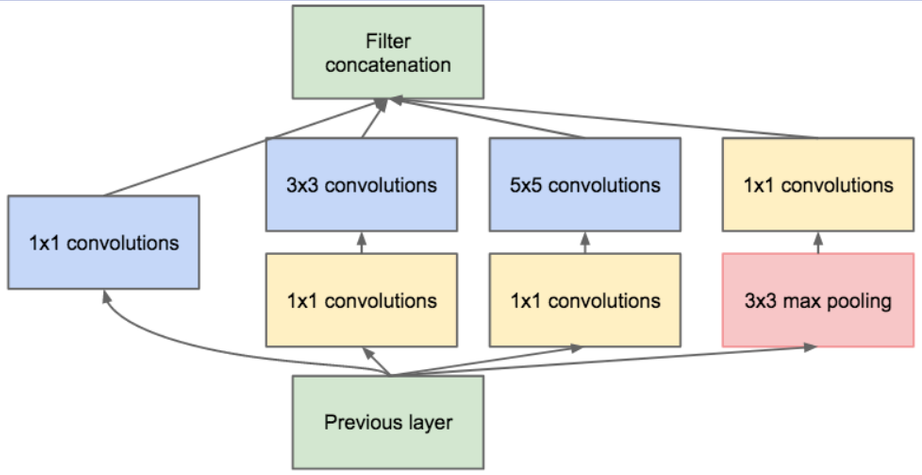
ILSVRC 2013 winner

Improvement on AlexNet by **tweaking the architecture hyperparameters**, in particular by expanding the size of the middle convolutional layers and making the stride and filter size on the first layer smaller.

Google LeNet (Inception Network)



(a) Inception module, naïve version



(b) Inception module with dimension reductions

Inception Module that dramatically reduces the number of parameters in the network (4M, compared to AlexNet with 60M)

Going deeper with convolutions

Christian Szegedy
Google Inc.

Wei Liu
University of North Carolina, Chapel Hill

Yangqing Jia
Google Inc.

Pierre Sermanet
Google Inc.

Scott Reed
University of Michigan

Dragomir Anguelov
Google Inc.

Dumitru Erhan
Google Inc.

Vincent Vanhoucke
Google Inc.

Andrew Rabinovich
Google Inc.

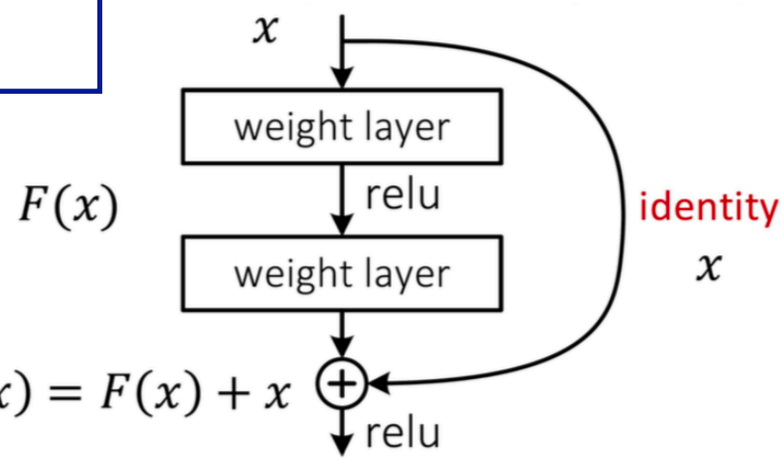
ILSVRC 2014 winner

ResNet

Deep Residual Learning for Image Recognition

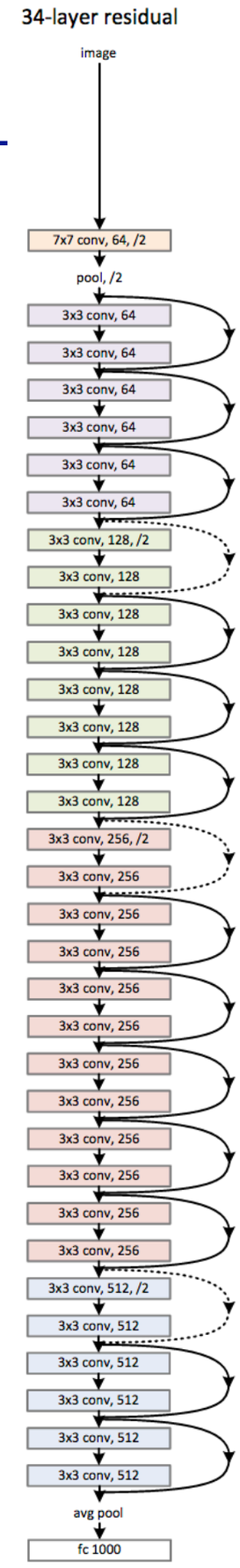
Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun
Microsoft Research
{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

This reformulation is motivated by the counterintuitive phenomena about the degradation problem (Fig. 1, left). As we discussed in the introduction, if the added layers can be constructed as identity mappings, a deeper model should have training error no greater than its shallower counterpart. The degradation problem suggests that the solvers might have difficulties in approximating identity mappings by multiple nonlinear layers. With the residual learning reformulation, if identity mappings are optimal, the solvers may simply drive the weights of the multiple nonlinear layers toward zero to approach identity mappings.

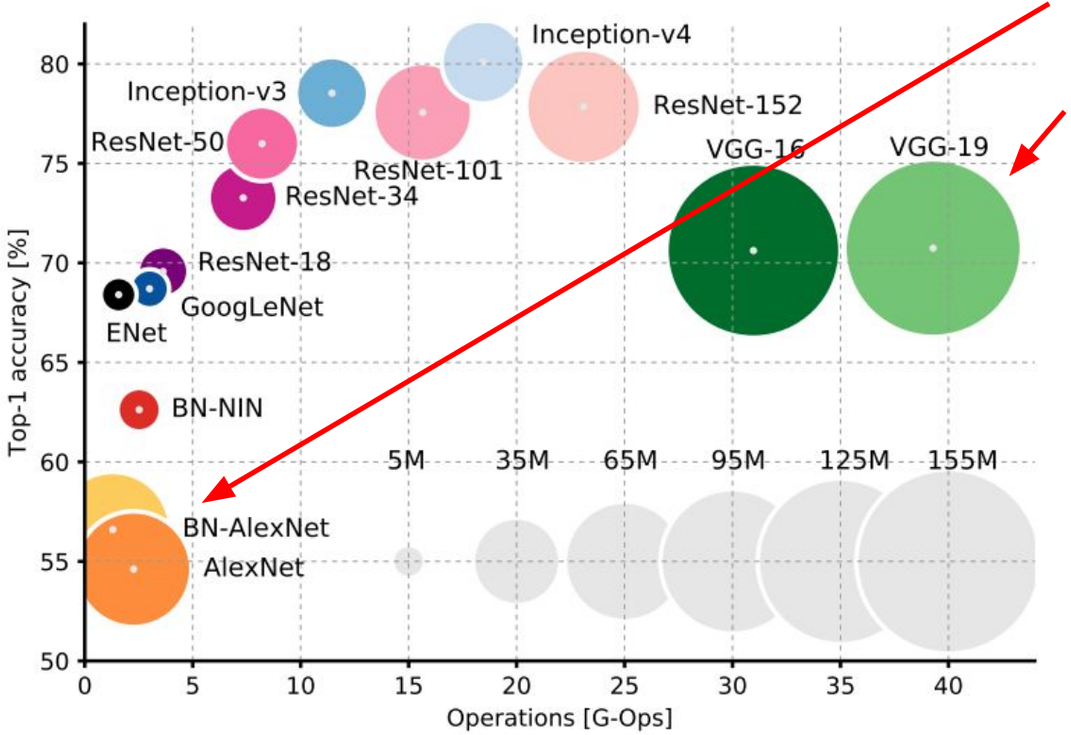


$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial H}\frac{\partial H}{\partial x} = \frac{\partial L}{\partial H}\left(\frac{\partial F}{\partial x} + 1\right)$$

- gradient skips weight layers – no vanishing
- improves gradient flow through layers



Overview



AlexNet and VGG have tons of parameters in the fully connected layers

AlexNet: ~62M parameters

FC6: 256x6x6 -> 4096: 38M params
FC7: 4096 -> 4096: 17M params
FC8: 4096 -> 1000: 4M params
~59M params in FC layers!

All provided in Pytorch torchvision module!

https://pytorch.org/docs/stable/torchvision/models.html