APS360 Fundamentals of AI

Lisa Zhang

Lecture 6; Jan 24, 2019

Agenda

- Convolutional Neural Networks
- Defining Convolutional Layers

Convolutional Neural Networks

Issue with Fully Connected Layers

- Input image: 200×200 pixels
- First hidden layer: 500 units

How many connections are there?

Issue with Fully Connected Layers

- Input image: 200×200 pixels
- First hidden layer: 500 units

How many connections are there?

Answer: 20 million!

Problematic because:

Issue with Fully Connected Layers

- Input image: 200×200 pixels
- First hidden layer: 500 units

How many connections are there?

Answer: 20 million!

Problematic because:

- computing predictions takes a long time
- a large number of weights requires a lot of training data to avoid overfitting
- small shift in image can result in large change in prediction

Convolutional Neural Network

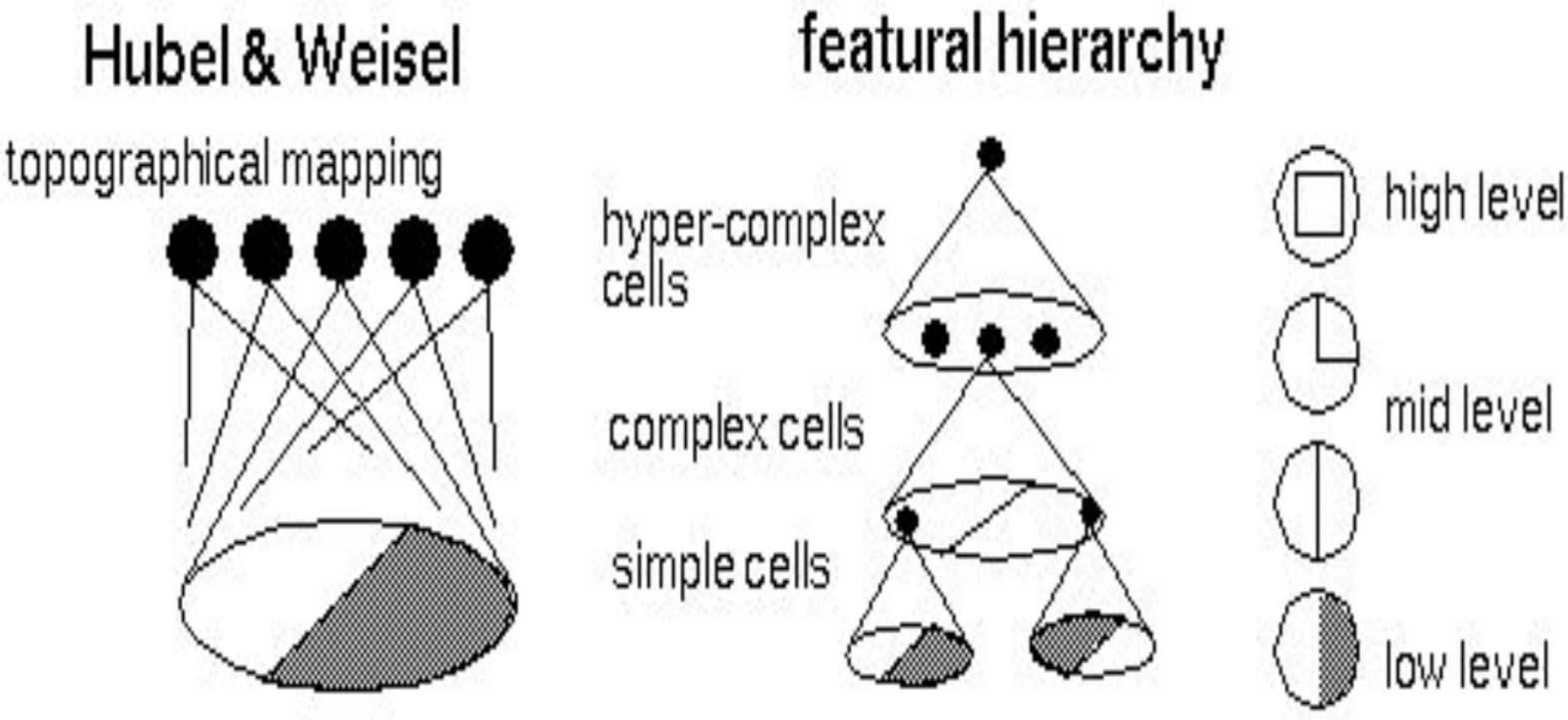
Ideas:

- Locally-connected layers: look for **local** features in small regions of the image
- Weight-sharing: detect the **same** local features across the entire image

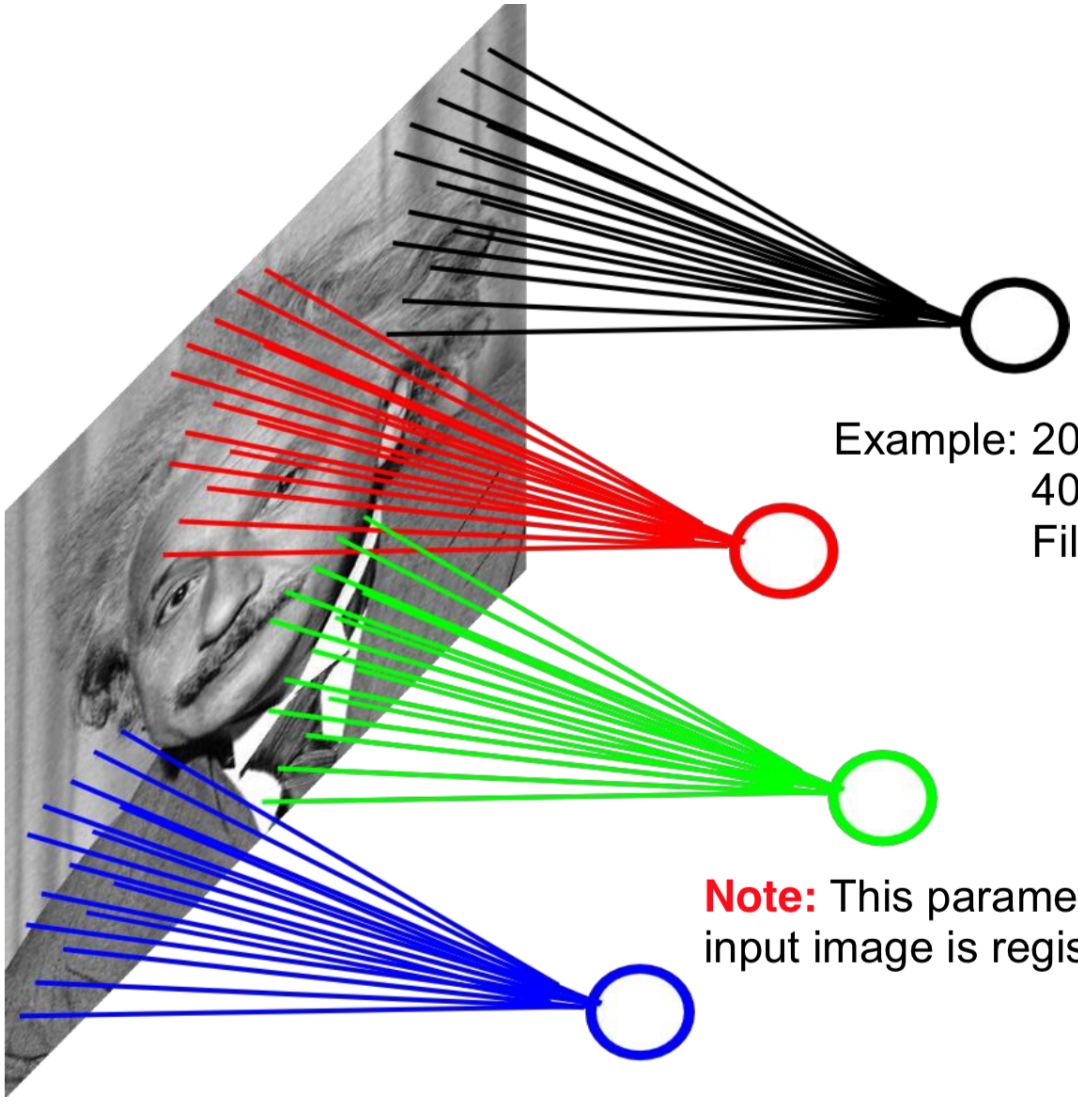
Convolutional Neural Network

Ideas:

- Locally-connected layers: look for **local** features in small regions of the image
- Weight-sharing: detect the **same** local features across the entire image



Locally Connected Layers

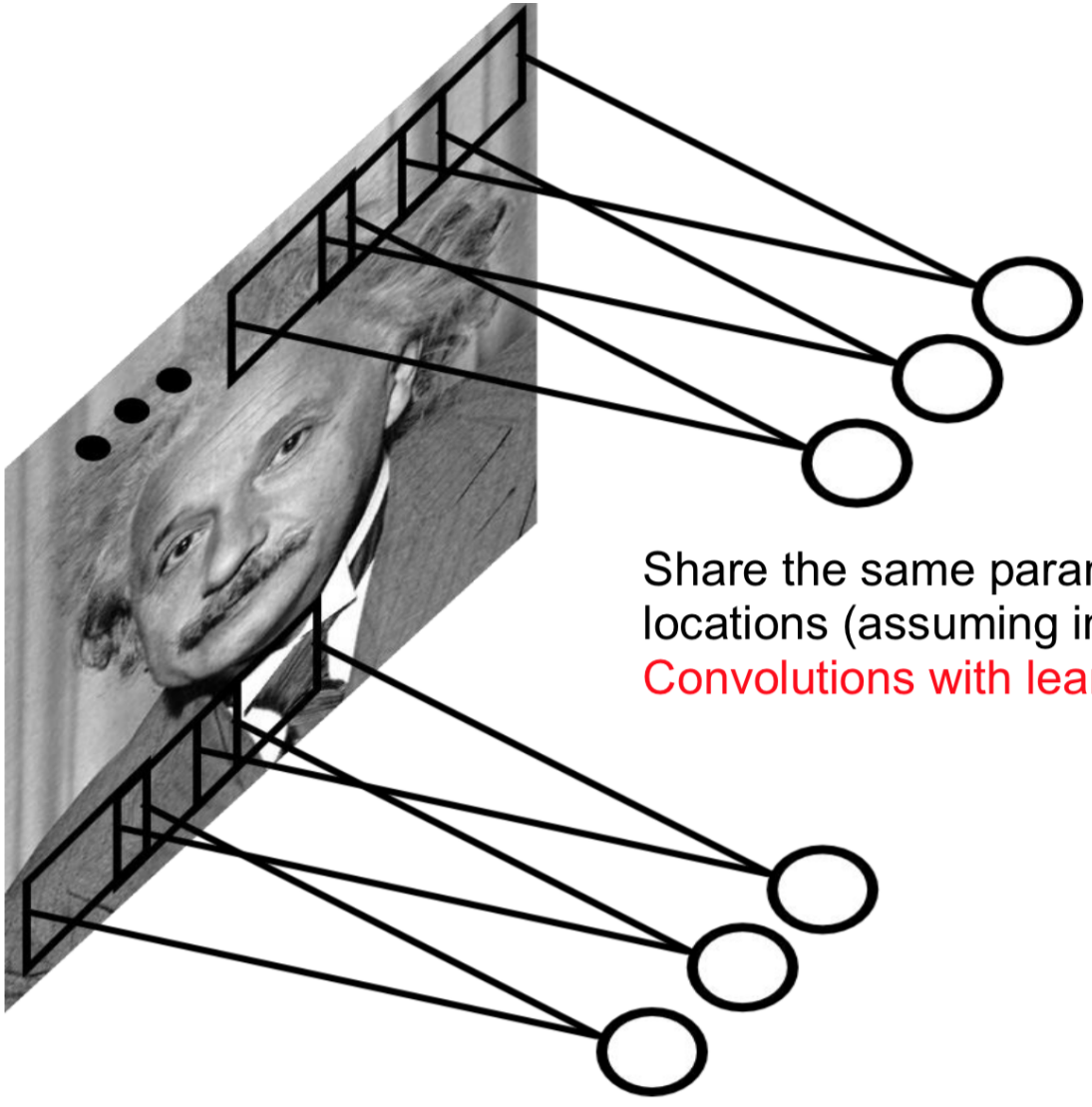


Example: 200x200 image
40K hidden units
Filter size: 10x10
4M parameters

Note: This parameterization is good when input image is registered (e.g., face recognition).

Ranzato

Weight Sharing



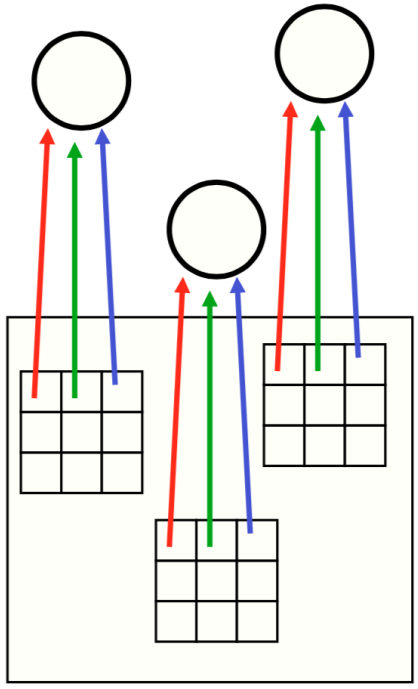
Share the same parameters across different locations (assuming input is stationary):
Convolutions with learned kernels

Ranzato

Weight Sharing (continued)

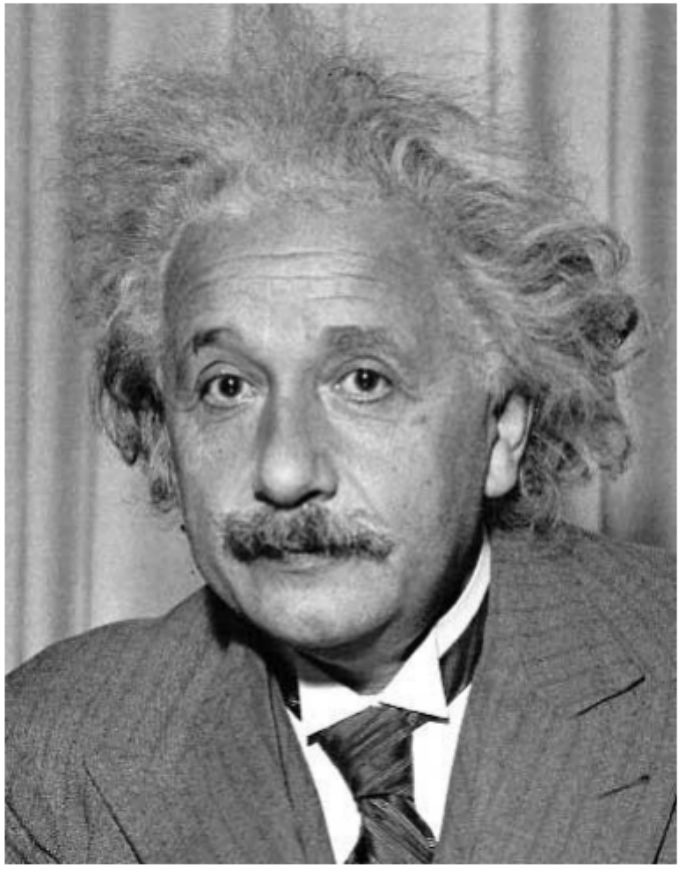
Idea: each neuron on the higher layer is detecting the same feature, but in different locations on the lower layer

The red connections all have the same weight.



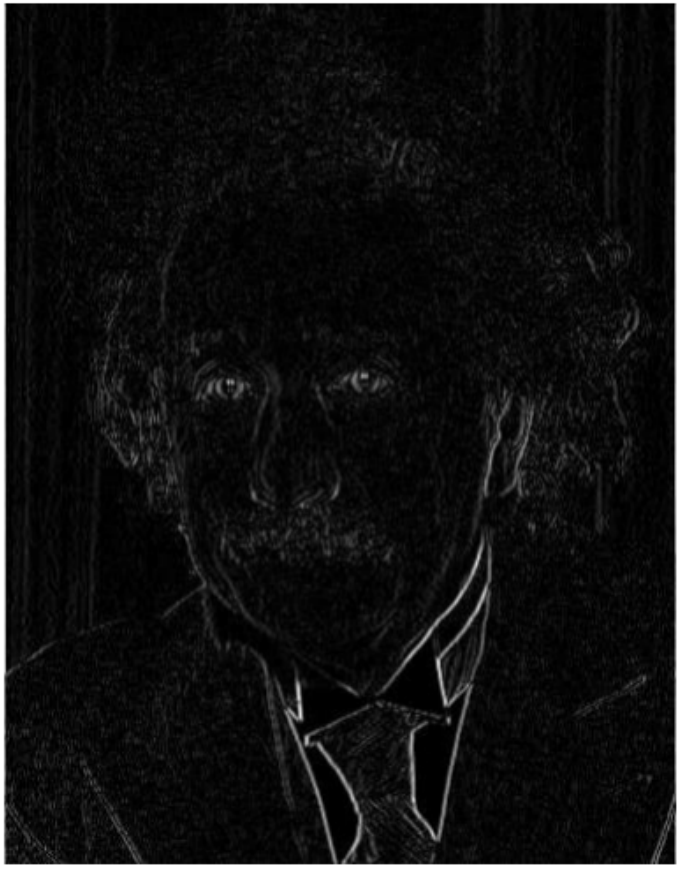
"Detecting" =the output (activation) is high if the feature is present

Sobel Filter - Weights to Detect Vertical Edges



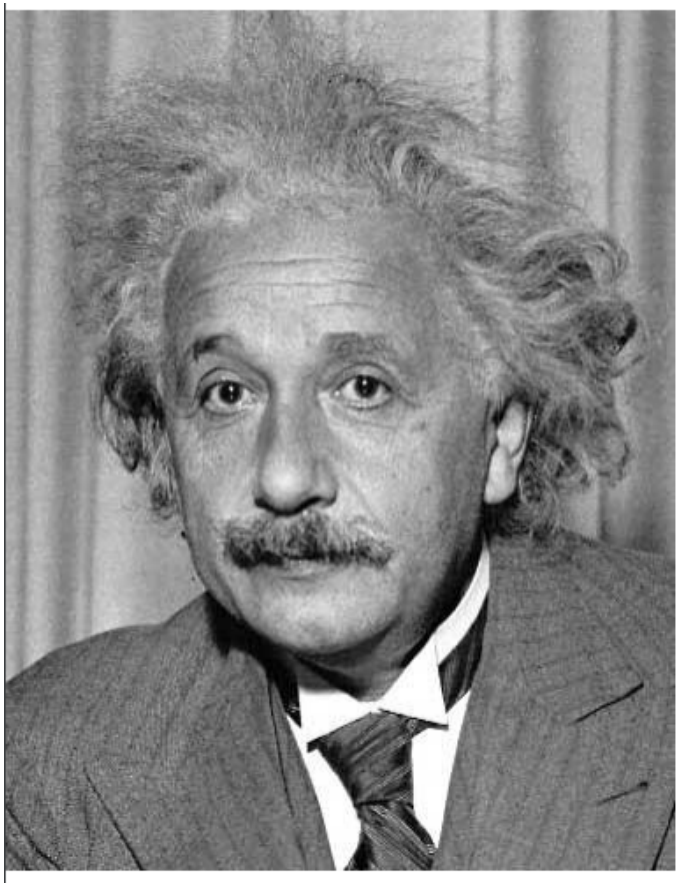
*

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



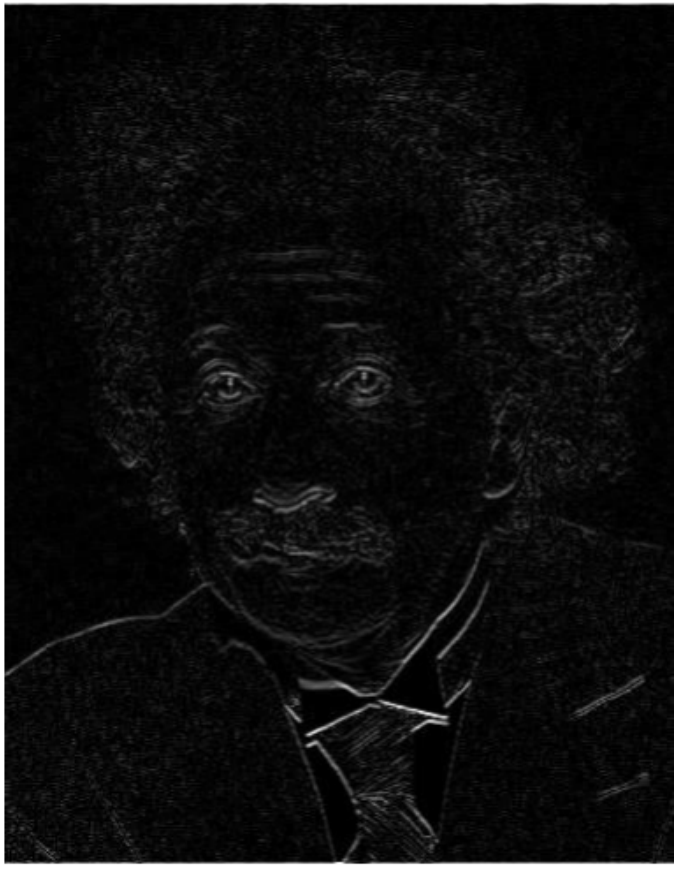
Vertical Edge
(absolute value)

Sobel Filter - Weights to Detect Horizontal Edges



*

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



Horizontal Edge
(absolute value)

Weights to Detect Blobs

$*$

$$\begin{pmatrix} 0 & 0 & 3 & 2 & 2 & 2 & 3 & 0 & 0 \\ 0 & 2 & 3 & 5 & 5 & 5 & 3 & 2 & 0 \\ 3 & 3 & 5 & 3 & 0 & 3 & 5 & 3 & 3 \\ 2 & 5 & 3 & -12 & -23 & -12 & 3 & 5 & 2 \\ 2 & 5 & 0 & -23 & -40 & -23 & 0 & 5 & 2 \\ 2 & 5 & 3 & -12 & -23 & -12 & 3 & 5 & 2 \\ 3 & 3 & 5 & 3 & 0 & 3 & 5 & 3 & 3 \\ 0 & 2 & 3 & 5 & 5 & 5 & 3 & 2 & 0 \\ 0 & 0 & 3 & 2 & 2 & 2 & 3 & 0 & 0 \end{pmatrix}$$

Example:

Greyscale input image of size 7×7 Convolution **kernel** of 3×3

Questions:

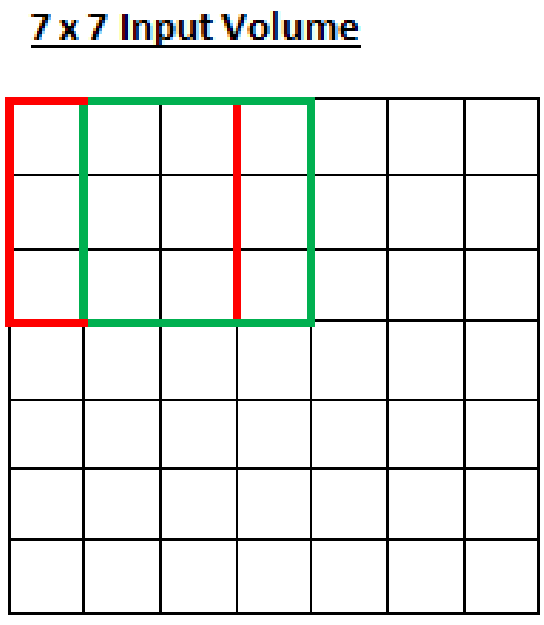
- How many units are in the higher layer?
- How many weights?

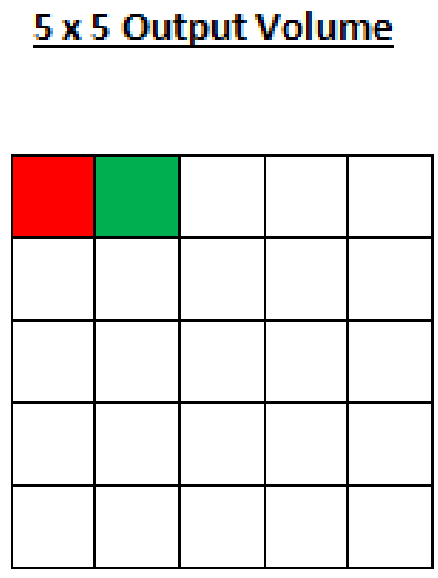
Example:

Greyscale input image of size 7×7 Convolution **kernel** of 3×3

Questions:

- How many units are in the higher layer?
- How many weights?





Computation:

http://deeplearning.stanford.edu/wiki/index.php/File:Convol

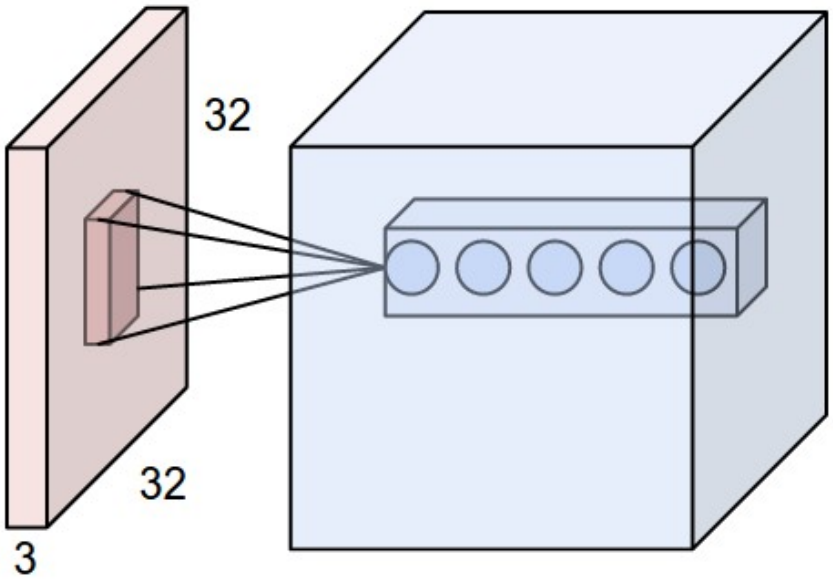
What if we have 3 colours?

Colour input image of size $7 \times 7 \times \mathbf{3}$ Convolution kernel of $3 \times 3 \times \mathbf{3}$

Questions:

- How many units are in the higher layer?
- How many weights?

Detecting Multiple Features



Input image of size $7 \times 7 \times 3$ Convolution kernel of $3 \times 3 \times 3 \times$ **5**

Questions:

- How many units are in the higher layer?
- How many weights?

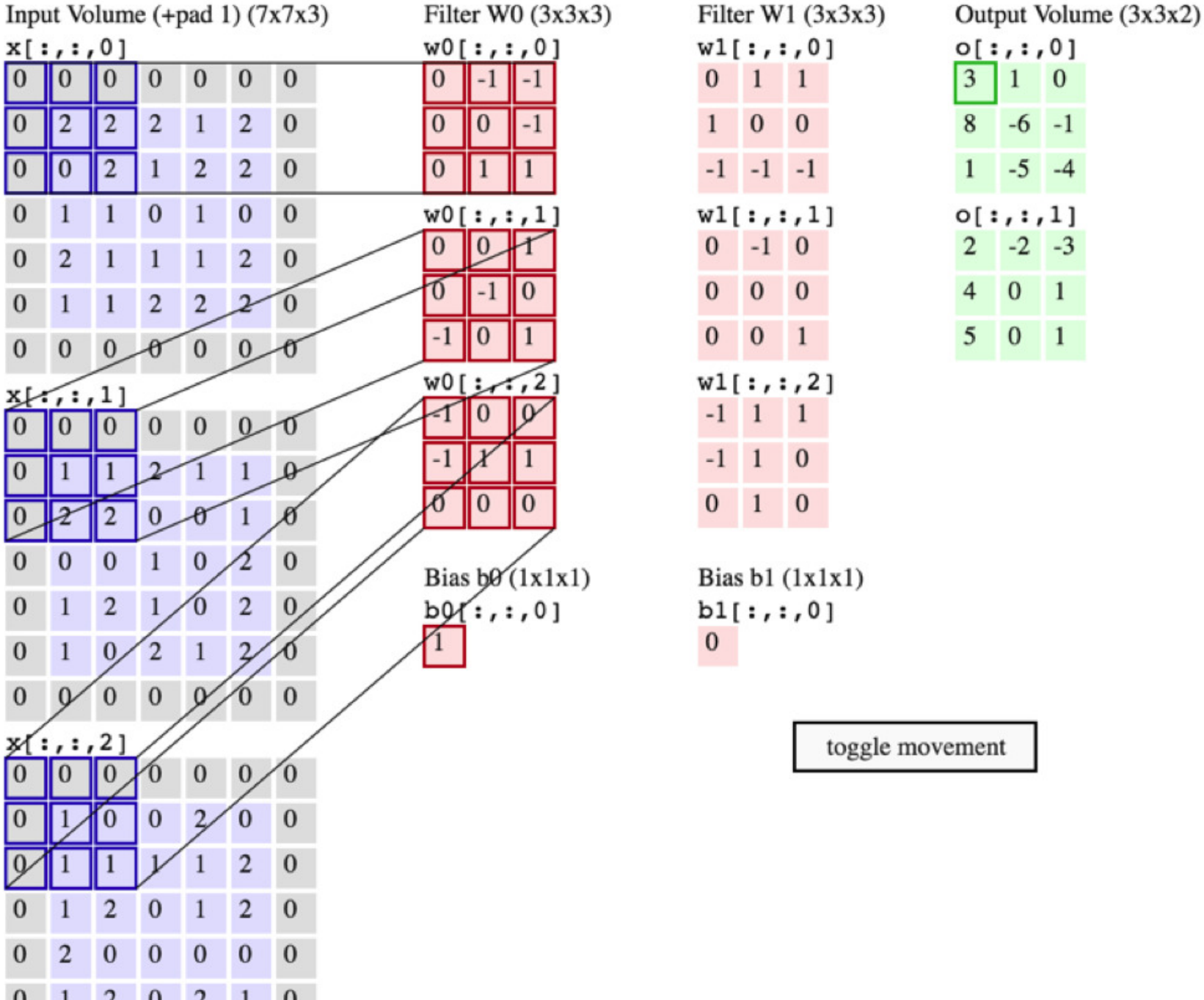
Example

Input image of size $32 \times 32 \times 3$ Convolution kernel of $3 \times 3 \times 3 \times \mathbf{10}$

Questions:

- How many units are in the higher layer?
- How many weights?

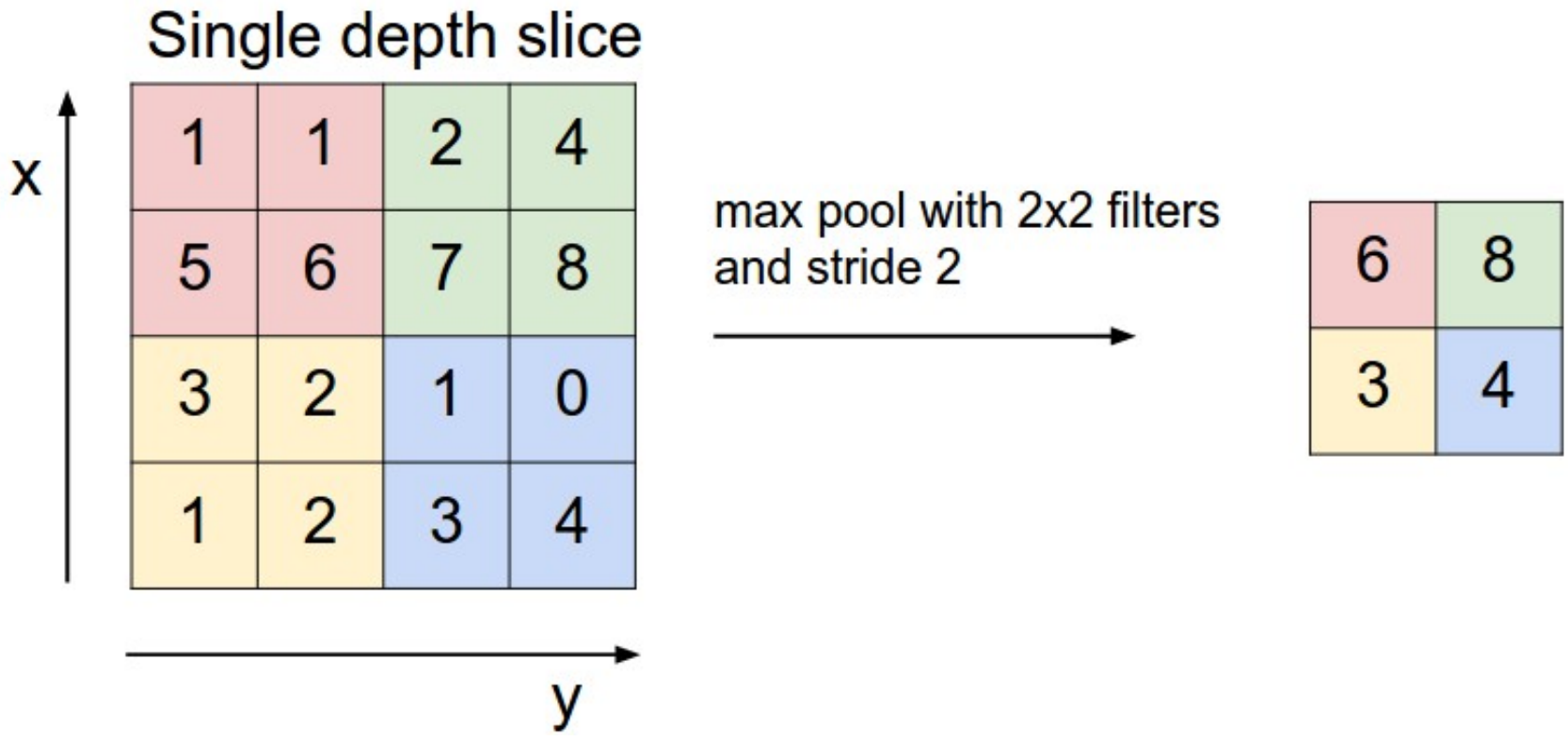
Zero Padding



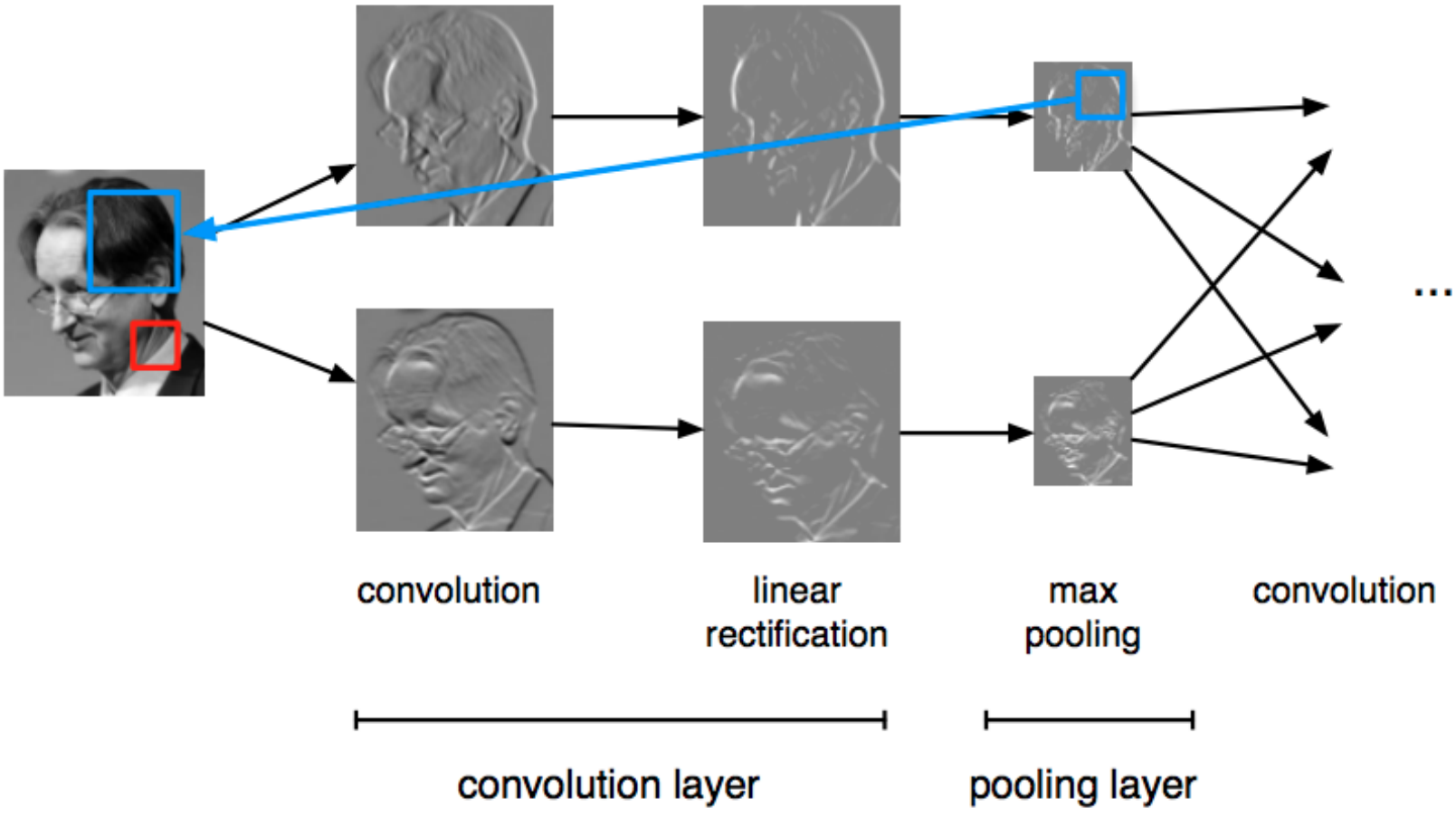
Why Zero Padding

- Keep the hidden unit “width” & “height” consistent, or at least still divisible by 2.
- Keep the information around the border of the image.

Max-Pooling


Single depth slice
x
y
1 1 2 4
5 6 7 8
3 2 1 0
1 2 3 4
max pool with 2x2 filters
and stride 2
6 8
3 4

Max-Pooling Example

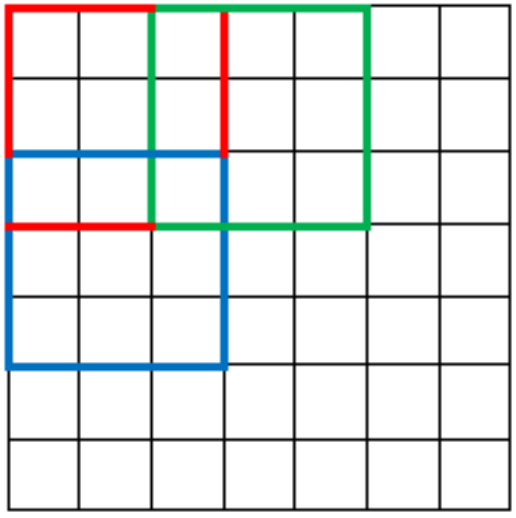


Other Types of Pooling Operations:

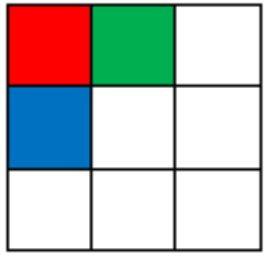
- Average pooling (compute the average activation of a region)
- Max pooling generally works better
- More recently people are doing away with pooling operations (and using strided convolutions instead)

Strided Convolution

7 x 7 Input Volume



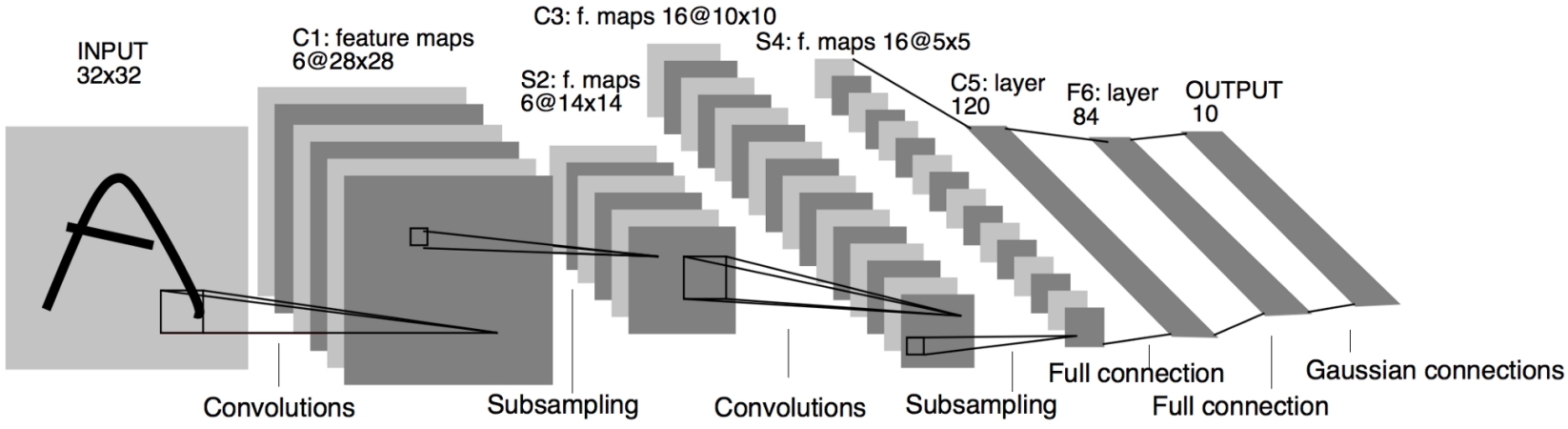
3 x 3 Output Volume



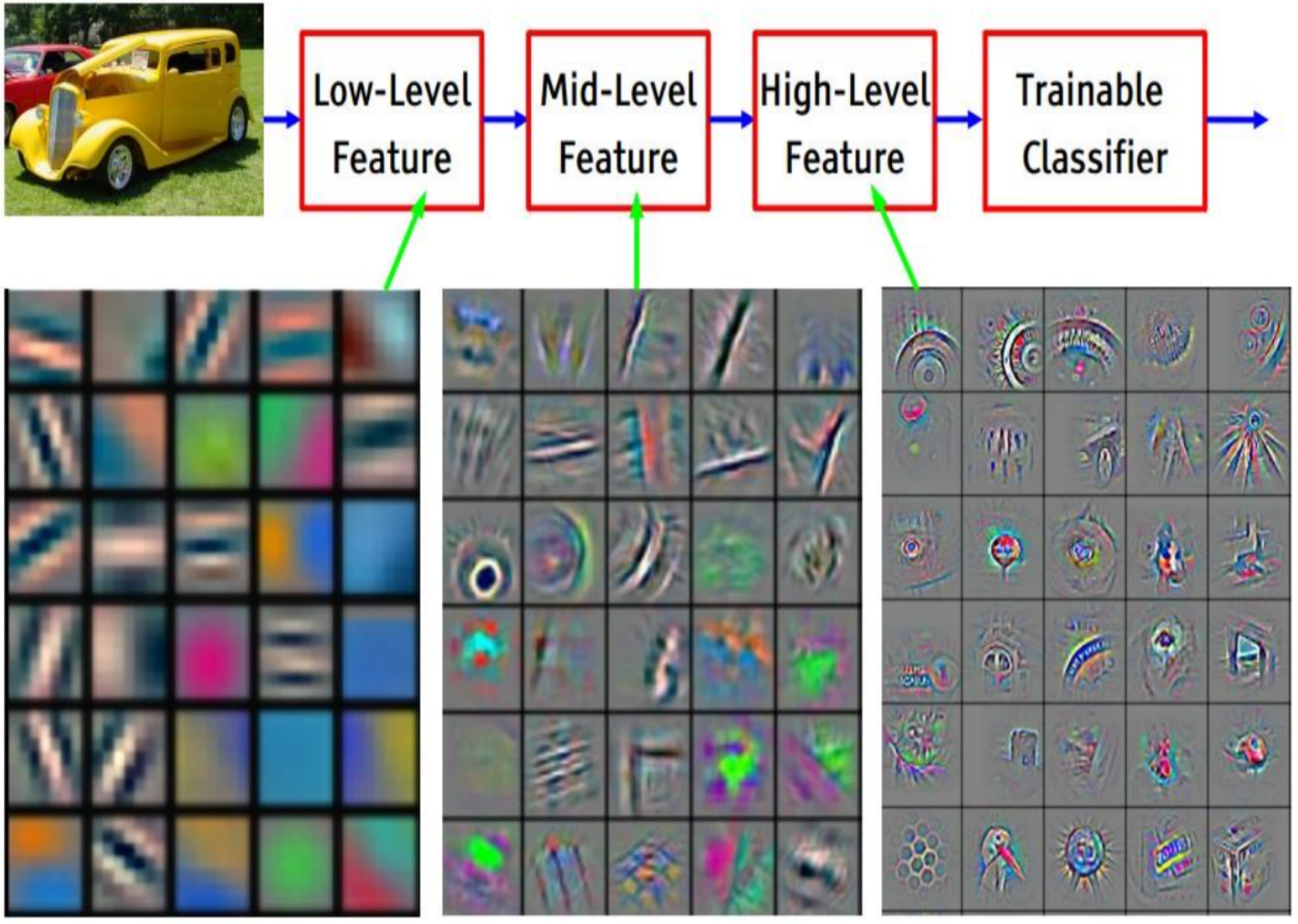
Animations

https://github.com/vdumoulin/conv_arithmetic

LeNet Architecture



What features do CNN's detect?



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

Convolutional Networks in PyTorch

Pytorch!