

---

**CS688: Web-Scale Image Search**  
**Convolutional Neural Networks**

---

**Sung-Eui Yoon**  
(윤성익)

**Course URL:**  
**<http://sgvr.kaist.ac.kr/~sungeui/IR>**

**KAIST**



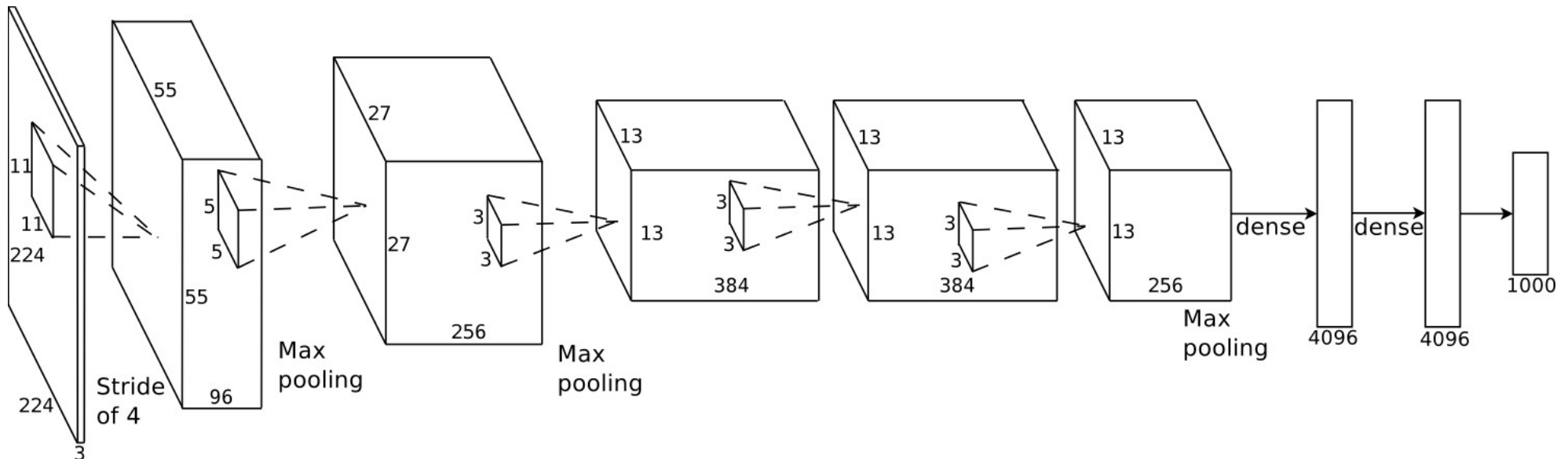
# Class Objectives

---

- **Review basics of convolution neural nets (CNNs)**
- **At the prior class:**
  - **Browsed main components of deep neural nets**

# Convolution Neural Nets (CNNs)

- **Deep neural nets, especially, CNNs, provide low-level and high-level features**
  - **We can use those features for image search**
- **Achieve the best results in many computer vision related problems**

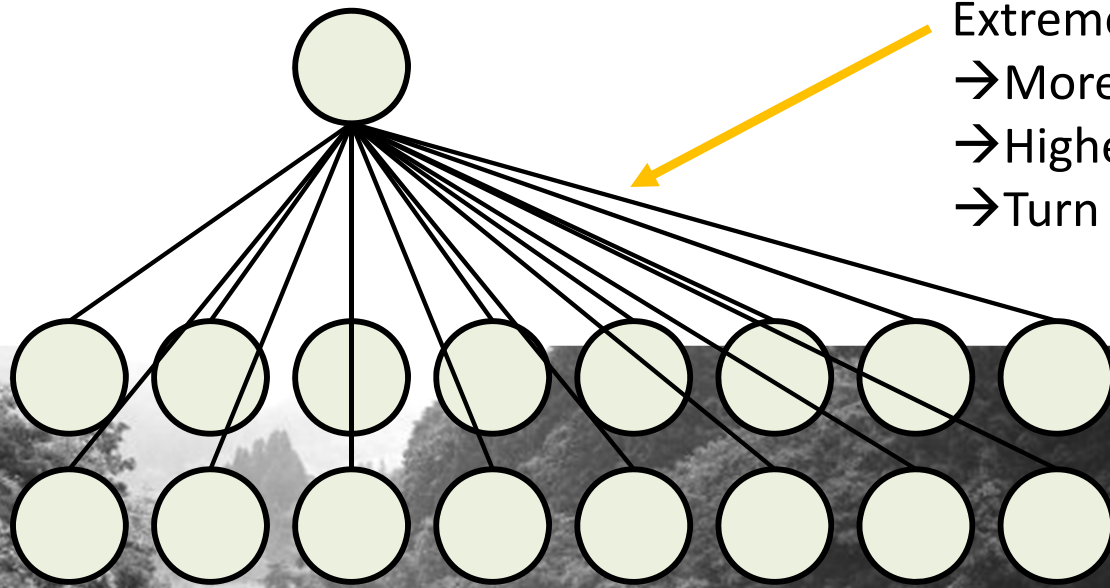


# Working with images

- Major factors for features:
  - Want to have “selective” and “invariant” features.
  - Try to exploit knowledge of images to accelerate training or improve performance.
- Generally try to avoid wiring detailed visual knowledge into system --- prefer to learn.

# Local connectivity

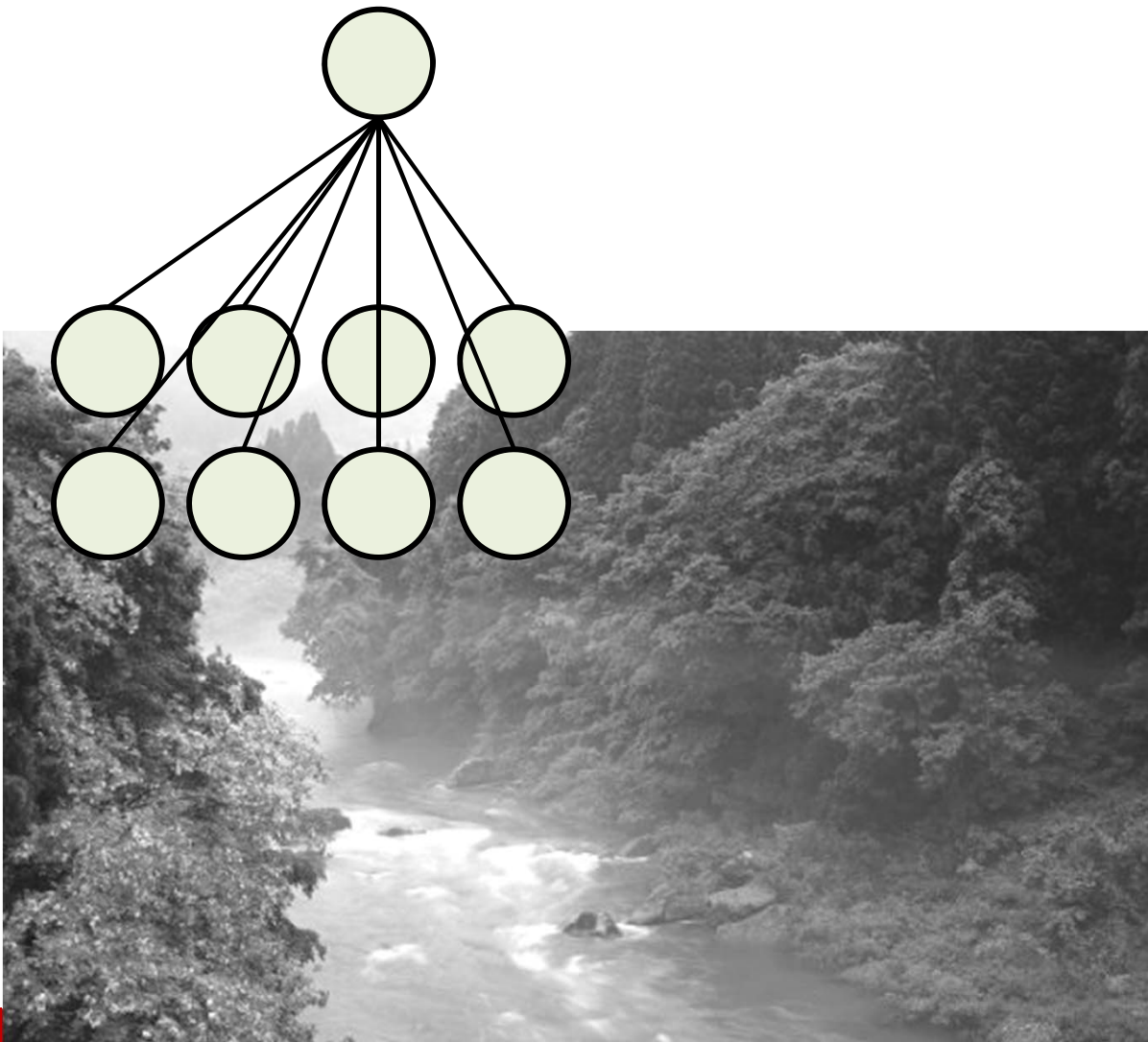
- Neural network view of single neuron:



- Extremely large number of connections.
- More parameters to train.
- Higher computational expense.
- Turn out not to be helpful in practice.

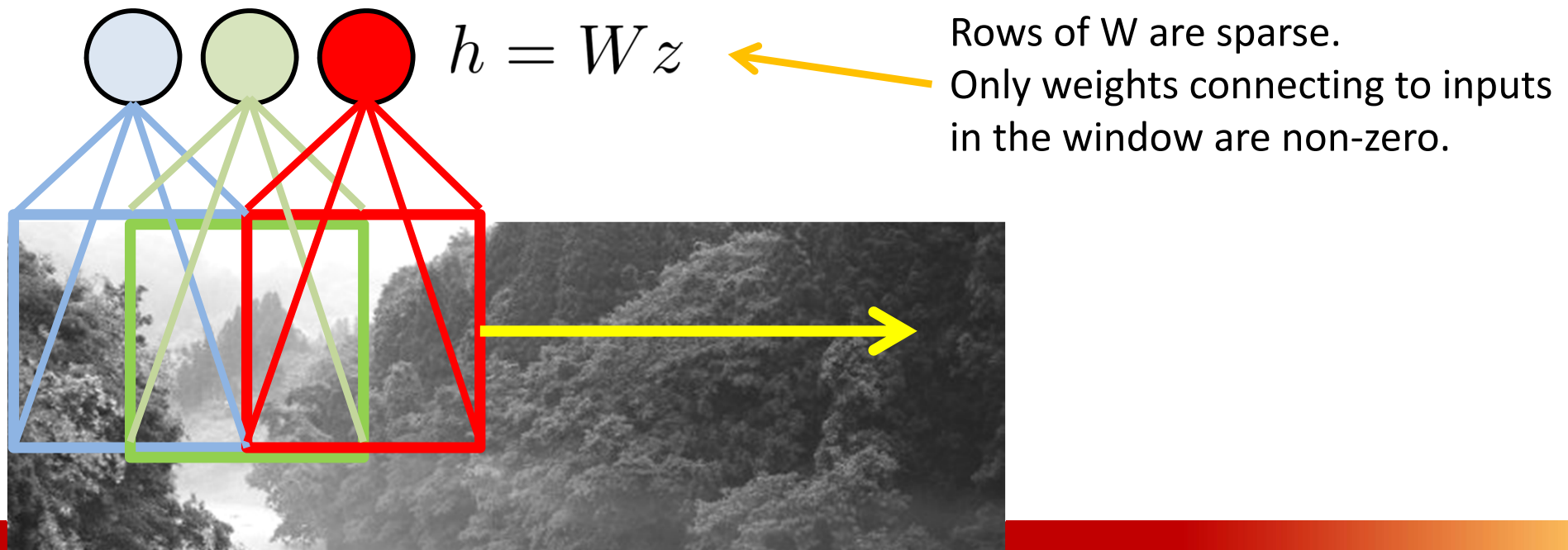
# Local connectivity

- Reduce parameters with local connections.
  - Weight vector is a spatially localized “filter”.



# Local connectivity

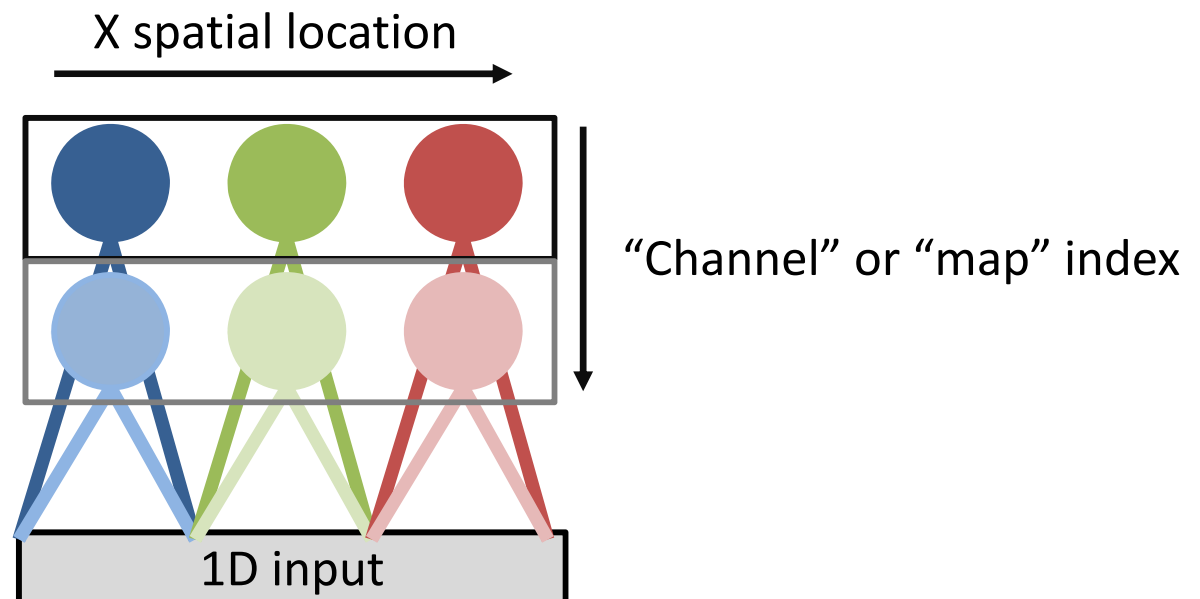
- Sometimes think of neurons as viewing small adjacent windows.
  - Specify connectivity by the size (“receptive field” size) and spacing (“step” or “stride”) of windows.
    - Typical RF size = 5 to 20
    - Typical step size = 1 pixel up to RF size.



# Local connectivity

- Spatial organization of filters means output features can also be organized like an image.
  - X,Y dimensions correspond to X,Y position of neuron window.
  - “Channels” are different features extracted from same spatial location. (Also called “feature maps”, or “maps”.)

1-dimensional example:

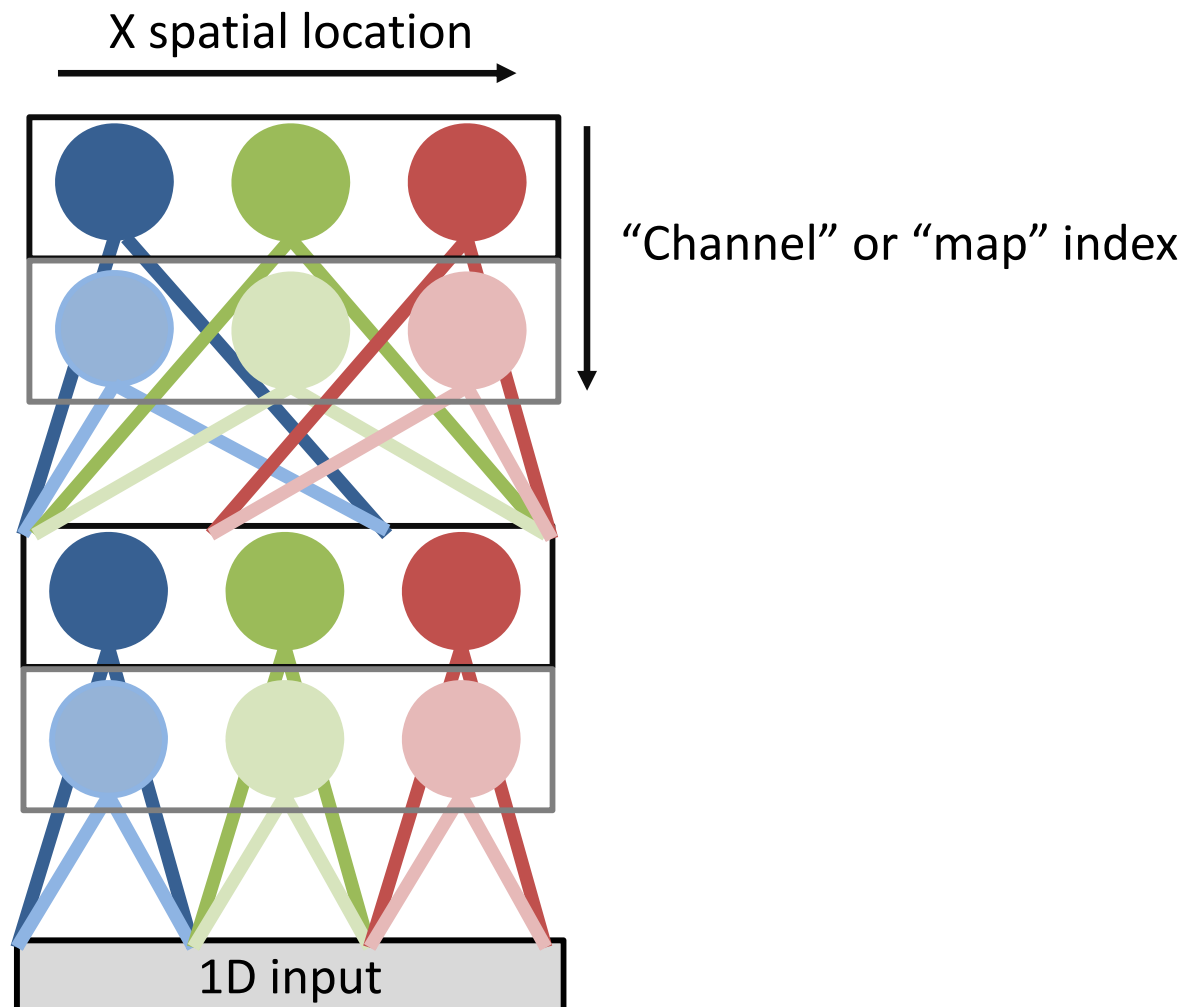




# Local connectivity

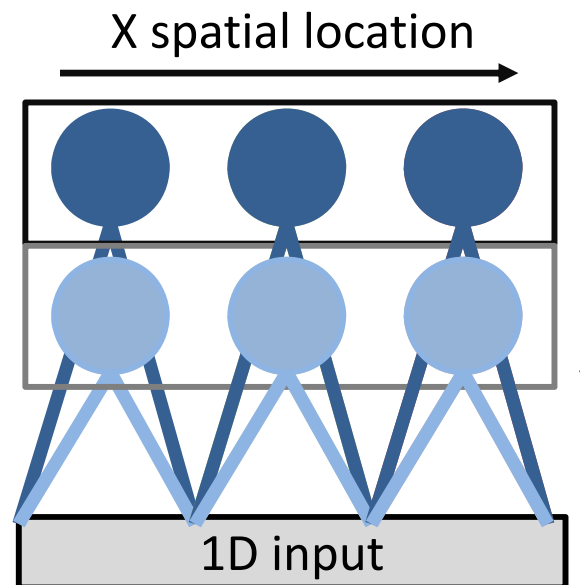
- We can treat output of a layer like an image and re-use the same tricks.

1-dimensional example:



# Weight-Tying or Convolutional Network

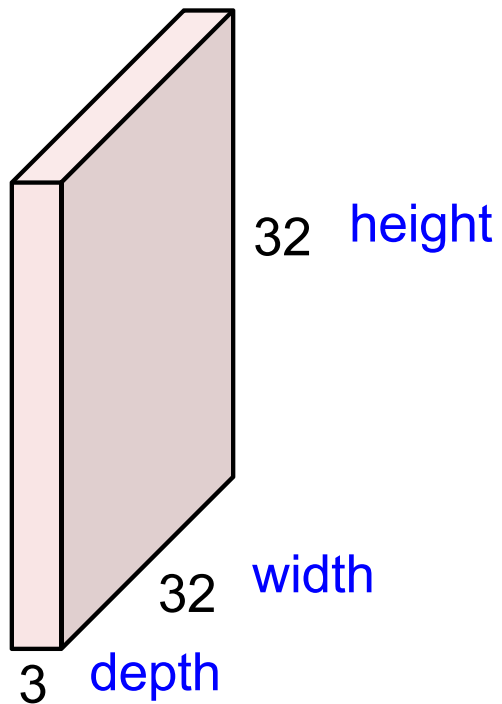
- Even with local connections, may still have too many weights.
  - Images tend to be “stationary”: different patches tend to have similar low-level structure.



- Each unique filter is spatially convolved with the input to produce responses for each map. [LeCun et al., 1989; LeCun et al., 2004]

# Convolution Layer

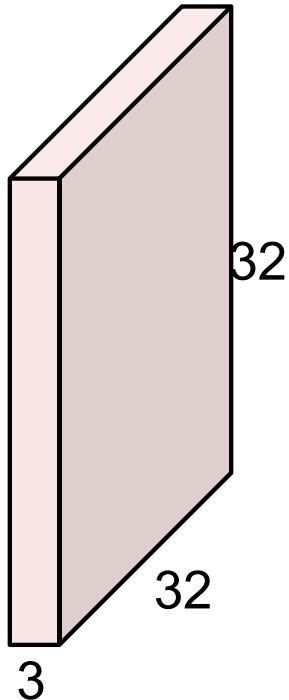
32x32x3 image -> preserve spatial structure



# Convolution Layer

Filters always extend the full depth of the input volume

32x32x3 image

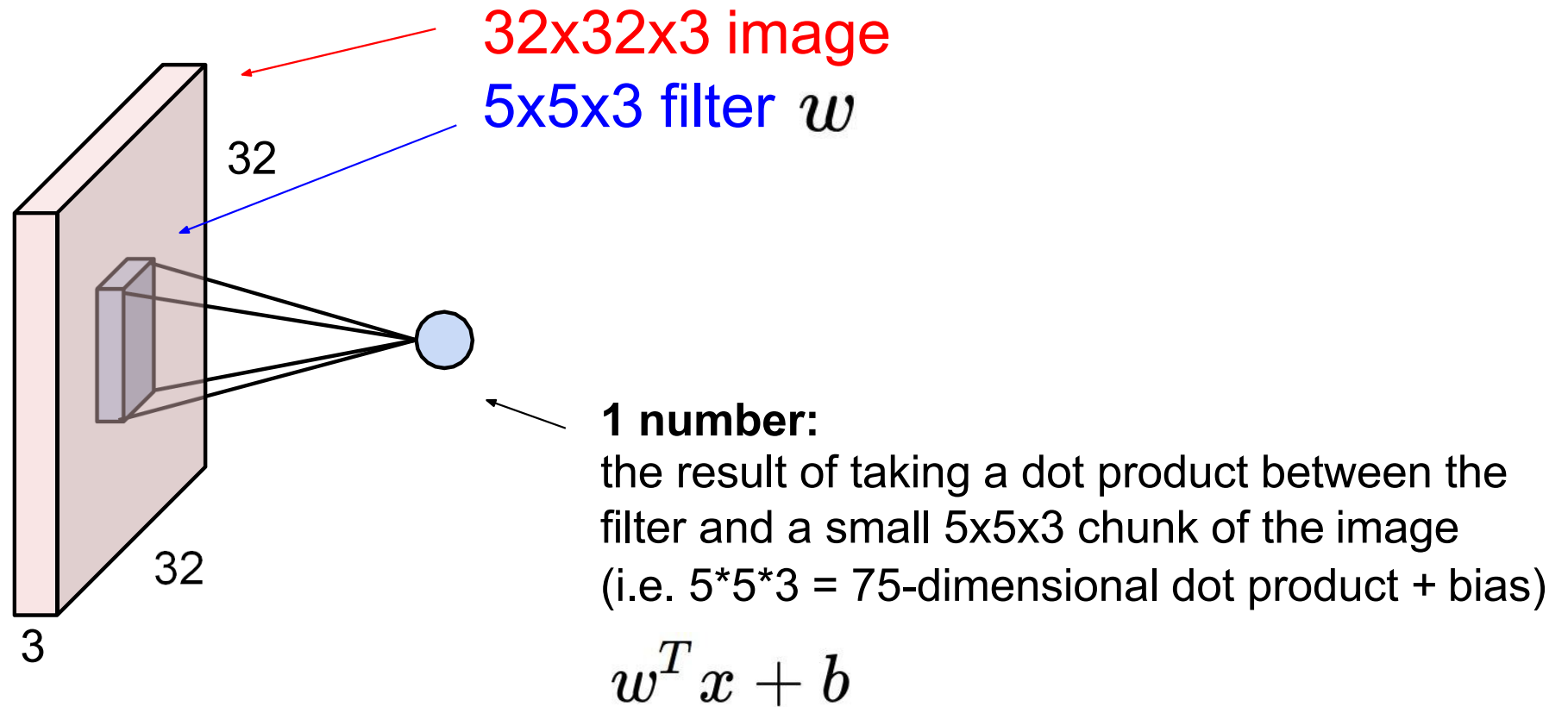


5x5x3 filter

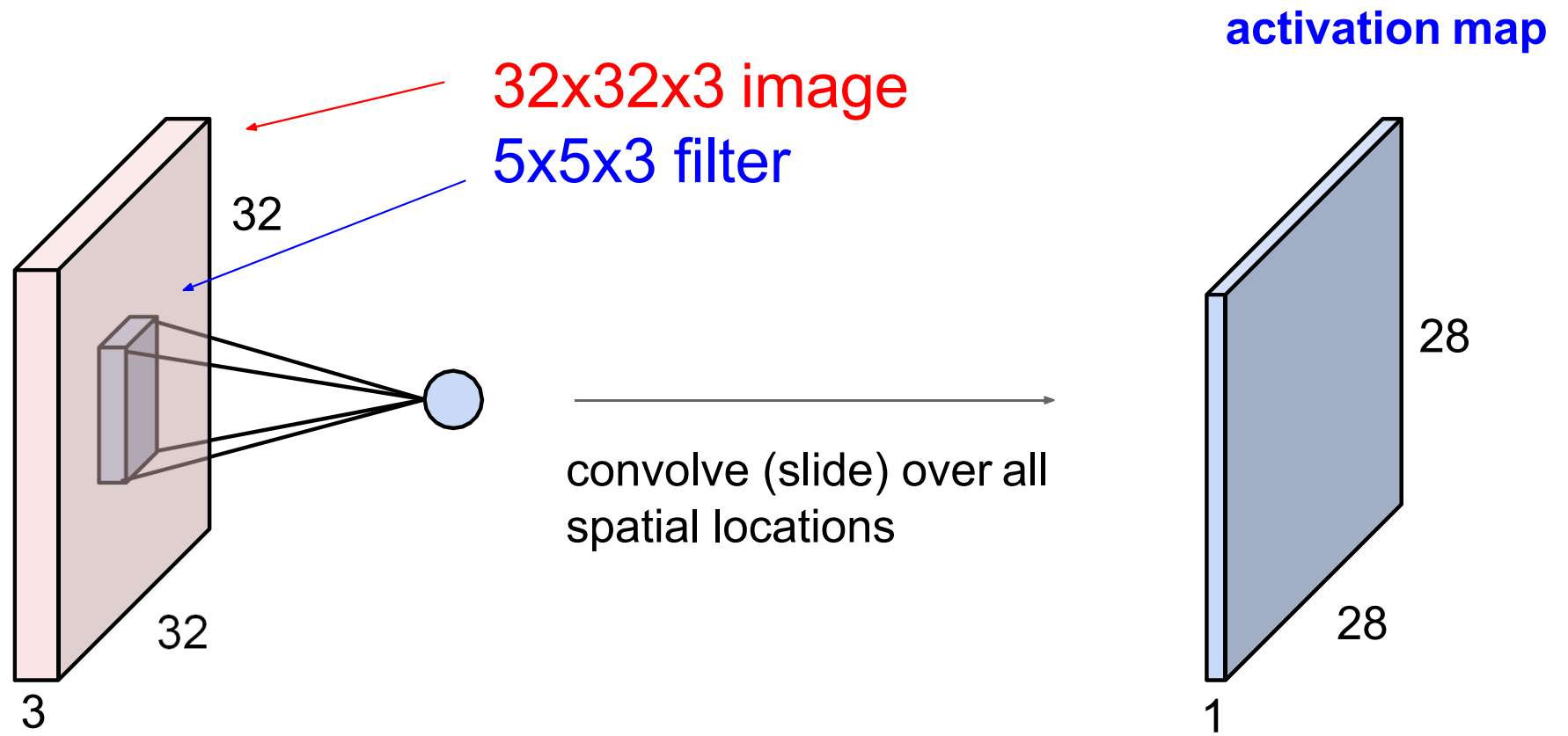


**Convolve** the filter with the image  
i.e. “slide over the image spatially,  
computing dot products”

# Convolution Layer

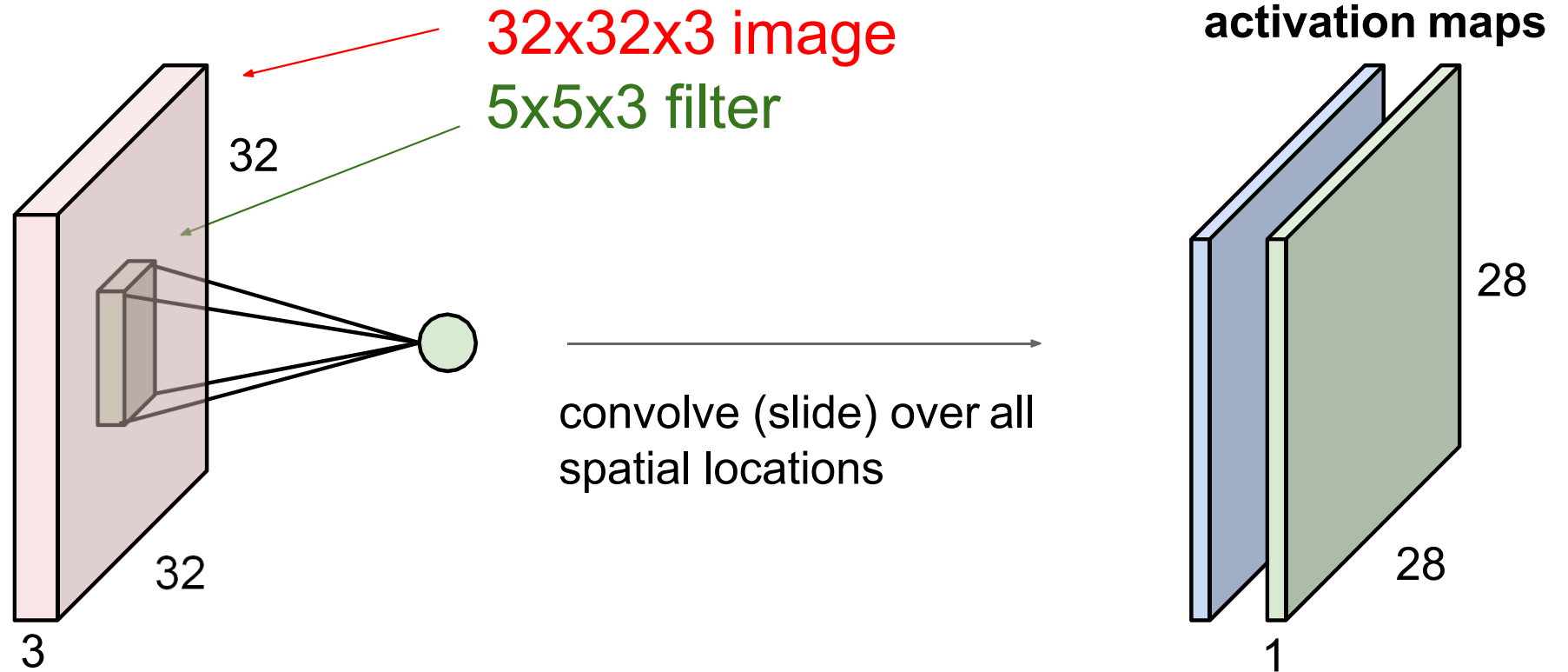


# Convolution Layer

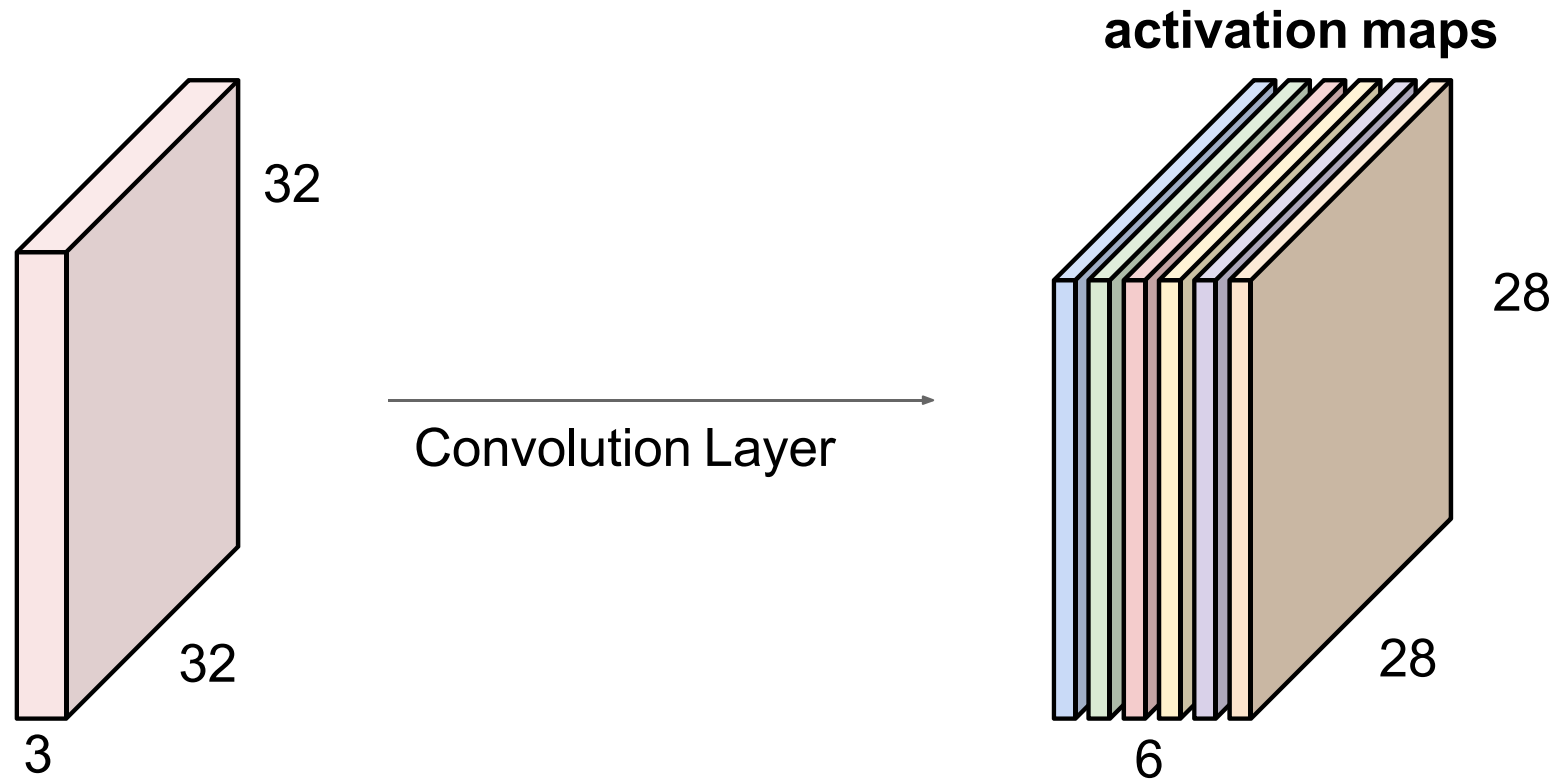


# Convolution Layer

consider a second, **green** filter



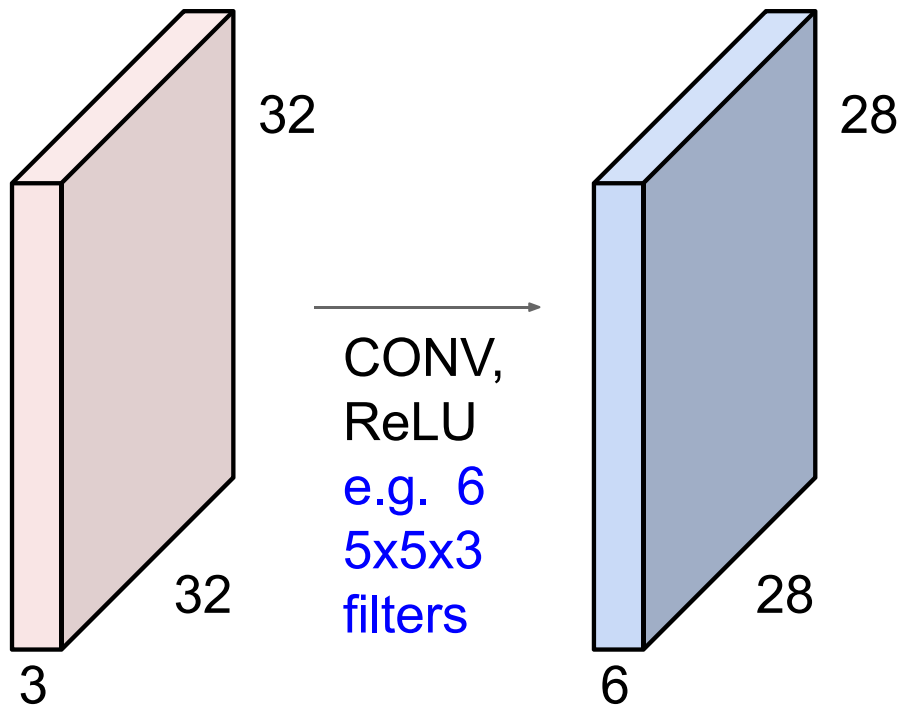
For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:



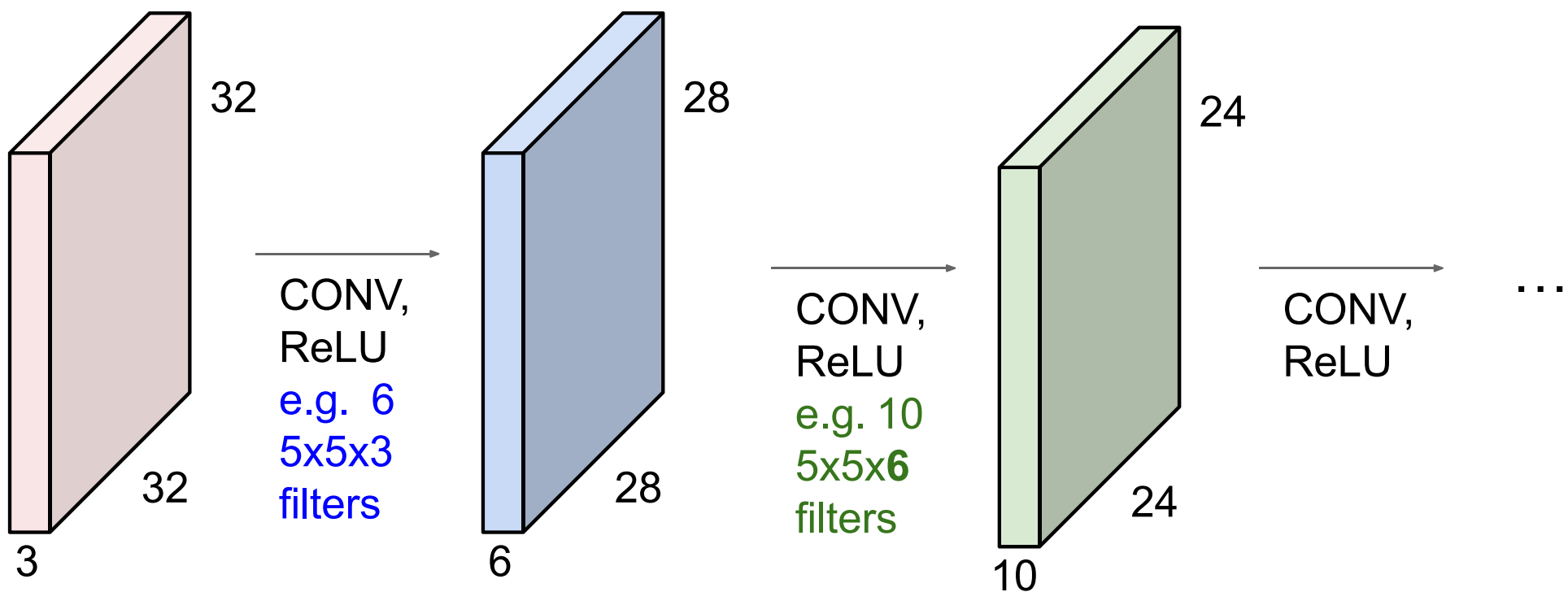
We stack these up to get a “new image” of size 28x28x6!



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



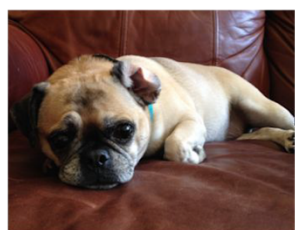
**Preview:** ConvNet is a sequence of Convolution Layers, interspersed with activation functions



# Preview

[Zeiler and Fergus 2013]

Visualization of VGG-16 by Lane McIntosh. VGG-16 architecture from [Simonyan and Zisserman 2014].

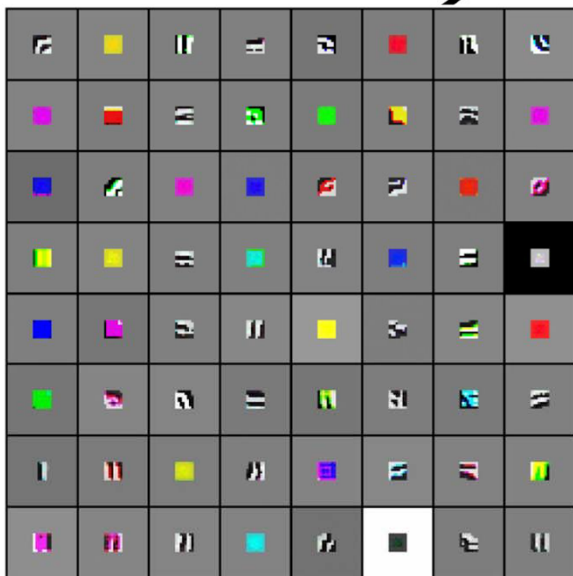


Low-level features

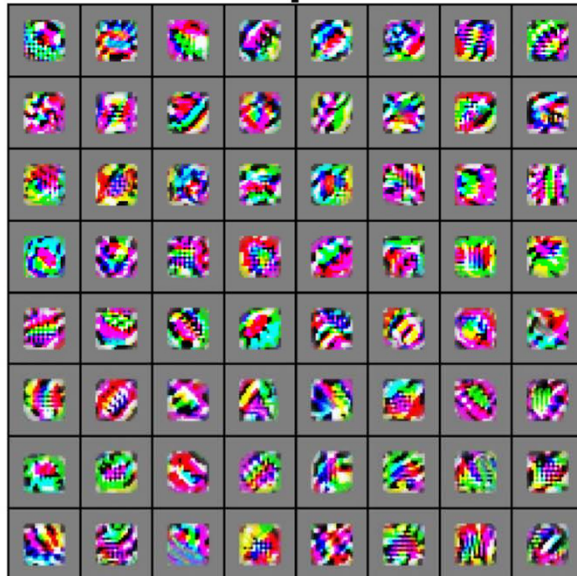
Mid-level features

High-level features

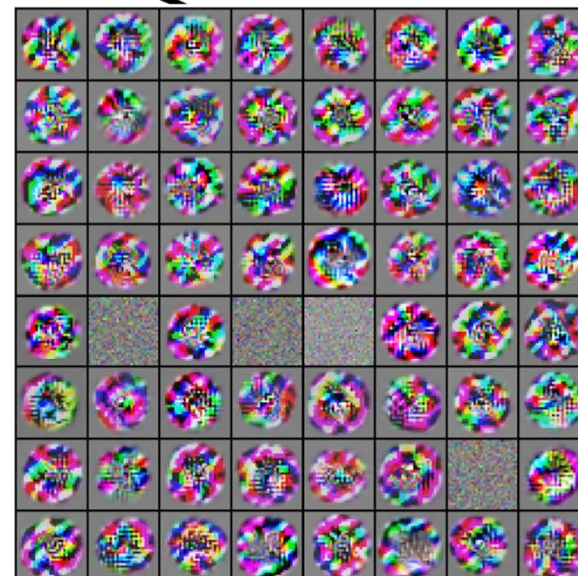
Linearly separable classifier



VGG-16 Conv1\_1



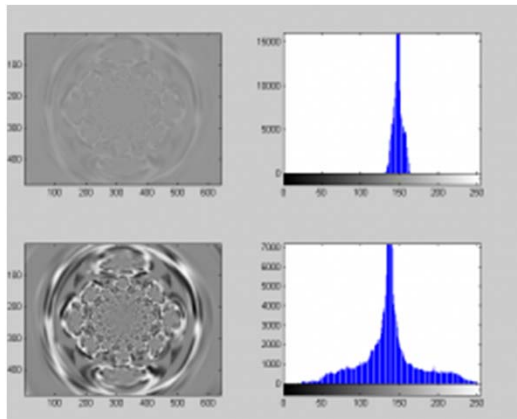
VGG-16 Conv3\_2



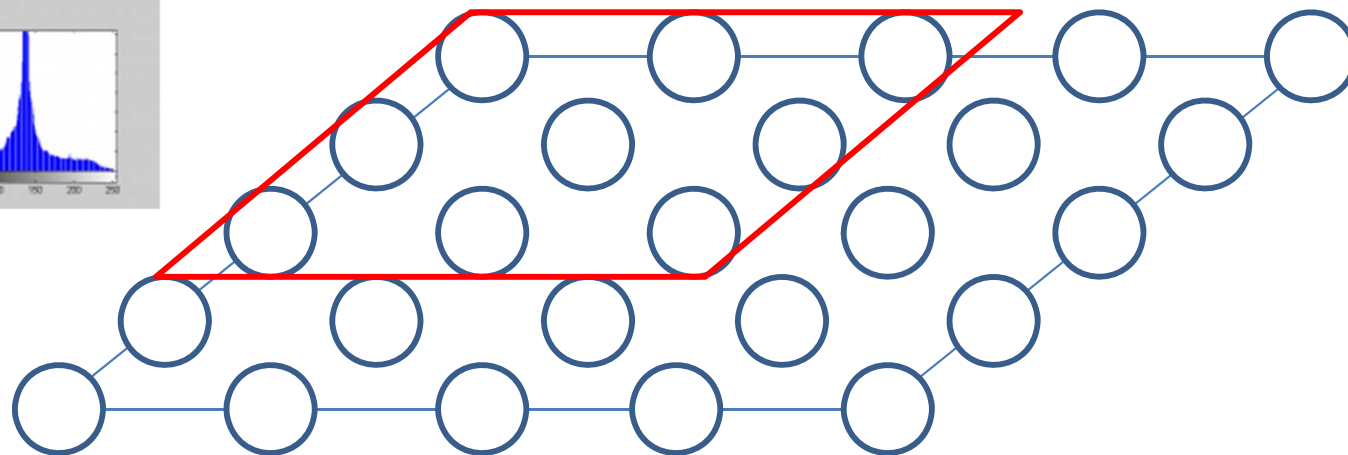
VGG-16 Conv5\_3

# Contrast Normalization

- Empirically useful to soft-normalize magnitude of groups of neurons.
  - Subtract out the local mean first.



Giassa.net

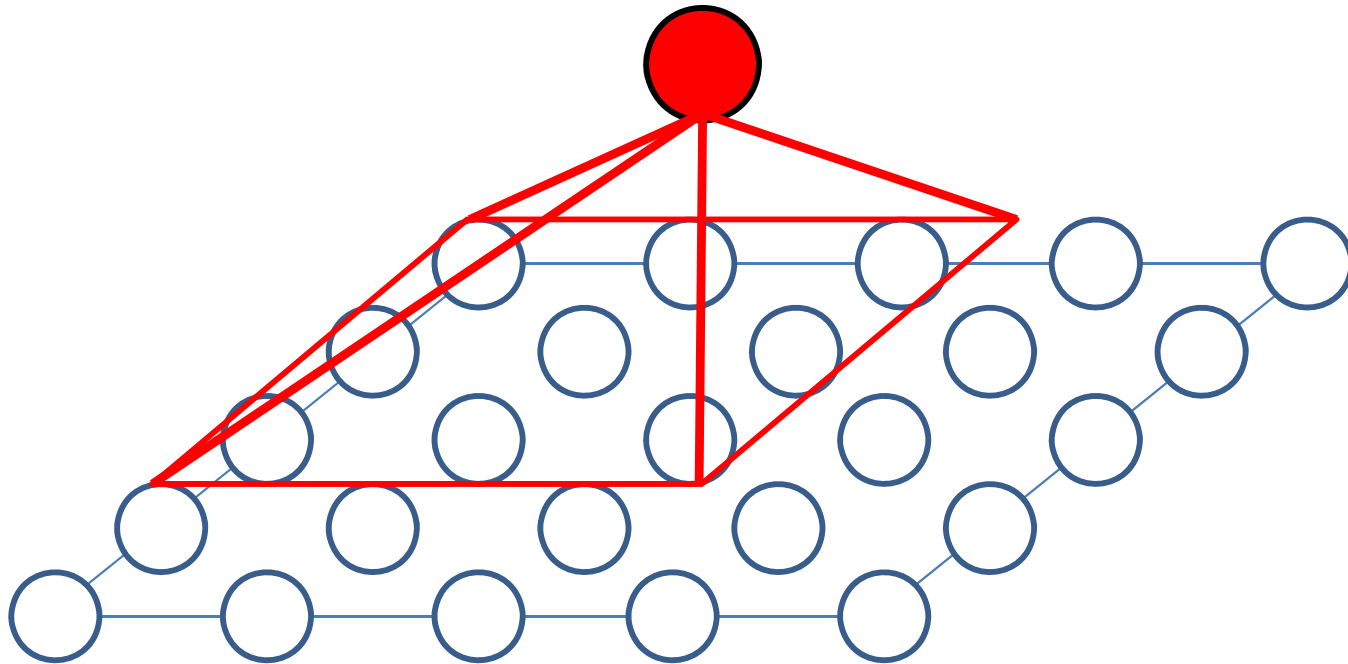


$$h = \frac{z}{\sqrt{W z^2 + \epsilon}}$$

→ **std.**

# Pooling

- Functional layers designed to represent invariant features.
  - Combined with convolution, corresponds to hard-wired translation invariance.
- Usually fix weights to local box or Gaussian filter.
  - Easy to represent max-, average-, or 2-norm pooling.

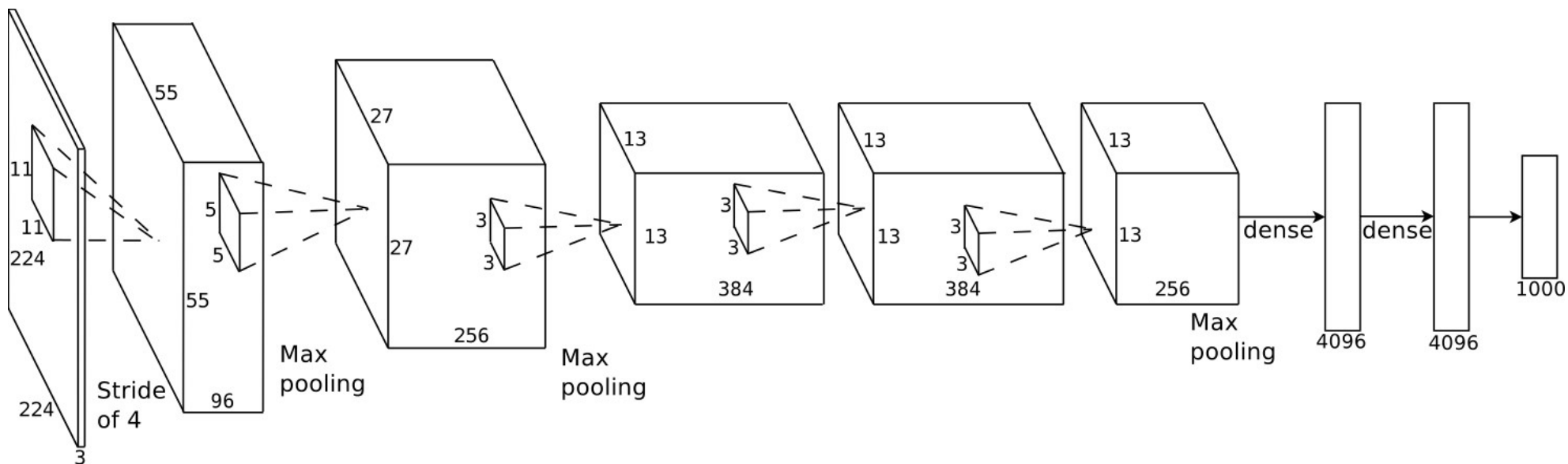


[Scherer et al., ICANN 2010]

[Boureau et al., ICML 2010]

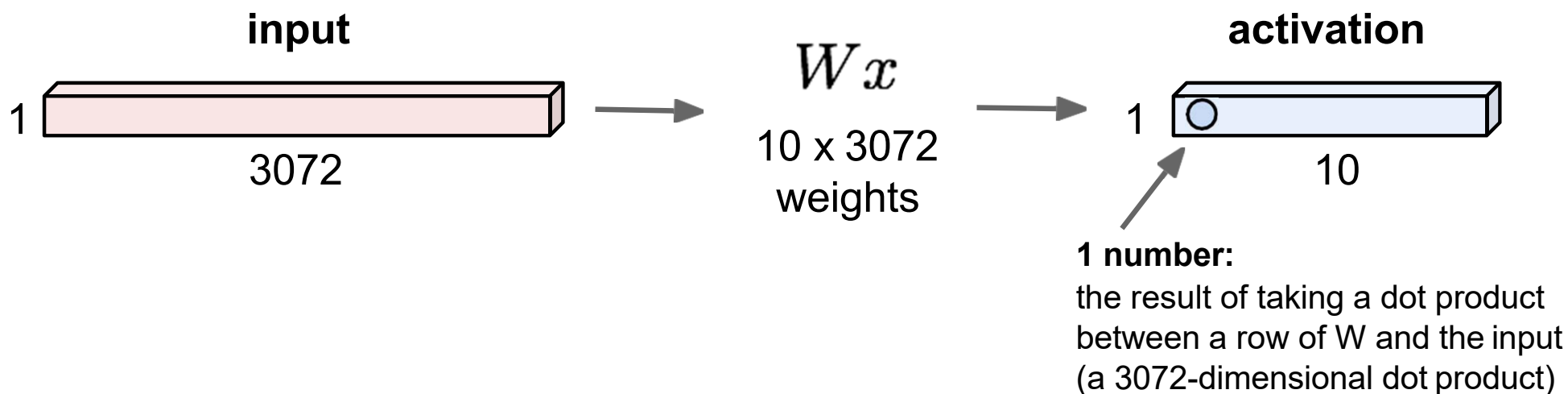
# Application: Image-Net

- System from [Krizhevsky et al., NIPS 2012](#):
  - Convolutional neural network.
  - Local connectivity.
  - Max-pooling.
  - Rectified linear units (ReLU).
  - Contrast normalization.











# Fully Connected Layer

32x32x3 image -> stretch to 3072 x 1



# Application: Image-Net

- Top result in LSVRC 2012: ~85%, Top-5 accuracy.

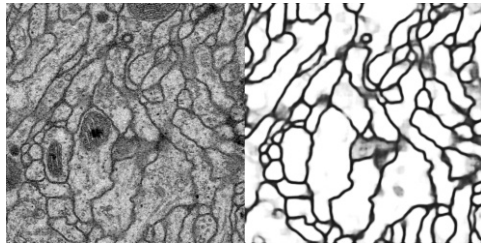
																							
<b>mite</b>	<b>container ship</b>	<b>motor scooter</b>																					
<table border="1"> <tbody> <tr><td>mite</td></tr> <tr><td>black widow</td></tr> <tr><td>cockroach</td></tr> <tr><td>tick</td></tr> <tr><td>starfish</td></tr> </tbody> </table>	mite	black widow	cockroach	tick	starfish	<table border="1"> <tbody> <tr><td>container ship</td></tr> <tr><td>lifeboat</td></tr> <tr><td>amphibian</td></tr> <tr><td>fireboat</td></tr> <tr><td>drilling platform</td></tr> </tbody> </table>	container ship	lifeboat	amphibian	fireboat	drilling platform	<table border="1"> <tbody> <tr><td>motor scooter</td></tr> <tr><td>go-kart</td></tr> <tr><td>moped</td></tr> <tr><td>bumper car</td></tr> <tr><td>golfcart</td></tr> </tbody> </table>	motor scooter	go-kart	moped	bumper car	golfcart						
mite																							
black widow																							
cockroach																							
tick																							
starfish																							
container ship																							
lifeboat																							
amphibian																							
fireboat																							
drilling platform																							
motor scooter																							
go-kart																							
moped																							
bumper car																							
golfcart																							
																							
<b>grille</b>	<b>mushroom</b>	<b>cherry</b>	<b>Ma</b>																				
<table border="1"> <tbody> <tr><td>convertible</td></tr> <tr><td>grille</td></tr> <tr><td>pickup</td></tr> <tr><td>beach wagon</td></tr> <tr><td>fire engine</td></tr> </tbody> </table>	convertible	grille	pickup	beach wagon	fire engine	<table border="1"> <tbody> <tr><td>agaric</td></tr> <tr><td>mushroom</td></tr> <tr><td>jelly fungus</td></tr> <tr><td>gill fungus</td></tr> <tr><td>dead-man's-fingers</td></tr> </tbody> </table>	agaric	mushroom	jelly fungus	gill fungus	dead-man's-fingers	<table border="1"> <tbody> <tr><td>dalmatian</td></tr> <tr><td>grape</td></tr> <tr><td>elderberry</td></tr> <tr><td>ffordshire bullterrier</td></tr> <tr><td>currant</td></tr> </tbody> </table>	dalmatian	grape	elderberry	ffordshire bullterrier	currant	<table border="1"> <tbody> <tr><td>squirrel monkey</td></tr> <tr><td>spider monkey</td></tr> <tr><td>titi</td></tr> <tr><td>indri</td></tr> <tr><td>howler monkey</td></tr> </tbody> </table>	squirrel monkey	spider monkey	titi	indri	howler monkey
convertible																							
grille																							
pickup																							
beach wagon																							
fire engine																							
agaric																							
mushroom																							
jelly fungus																							
gill fungus																							
dead-man's-fingers																							
dalmatian																							
grape																							
elderberry																							
ffordshire bullterrier																							
currant																							
squirrel monkey																							
spider monkey																							
titi																							
indri																							
howler monkey																							

What's an Agaric!?



# More applications

- Segmentation: predict classes of pixels / super-pixels.



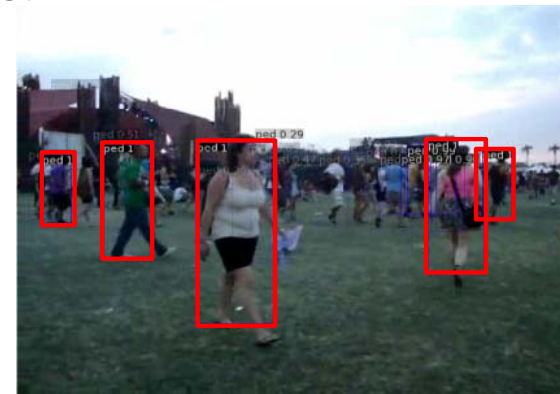
Farabet et al., ICML 2012 →



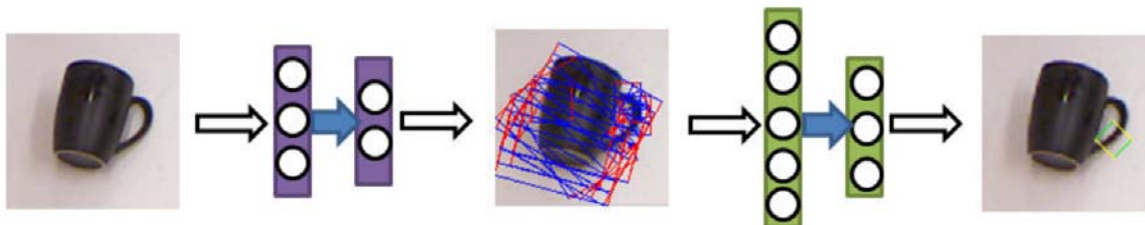
← Ciresan et al., NIPS 2012

- Detection: combine classifiers with sliding-window architecture.
  - Economical when used with convolutional nets.

Pierre Sermanet (2010) →



- Robotic grasping. [Lenz et al., RSS 2013]



<http://www.youtube.com/watch?v=f9Cuzql1SkE>

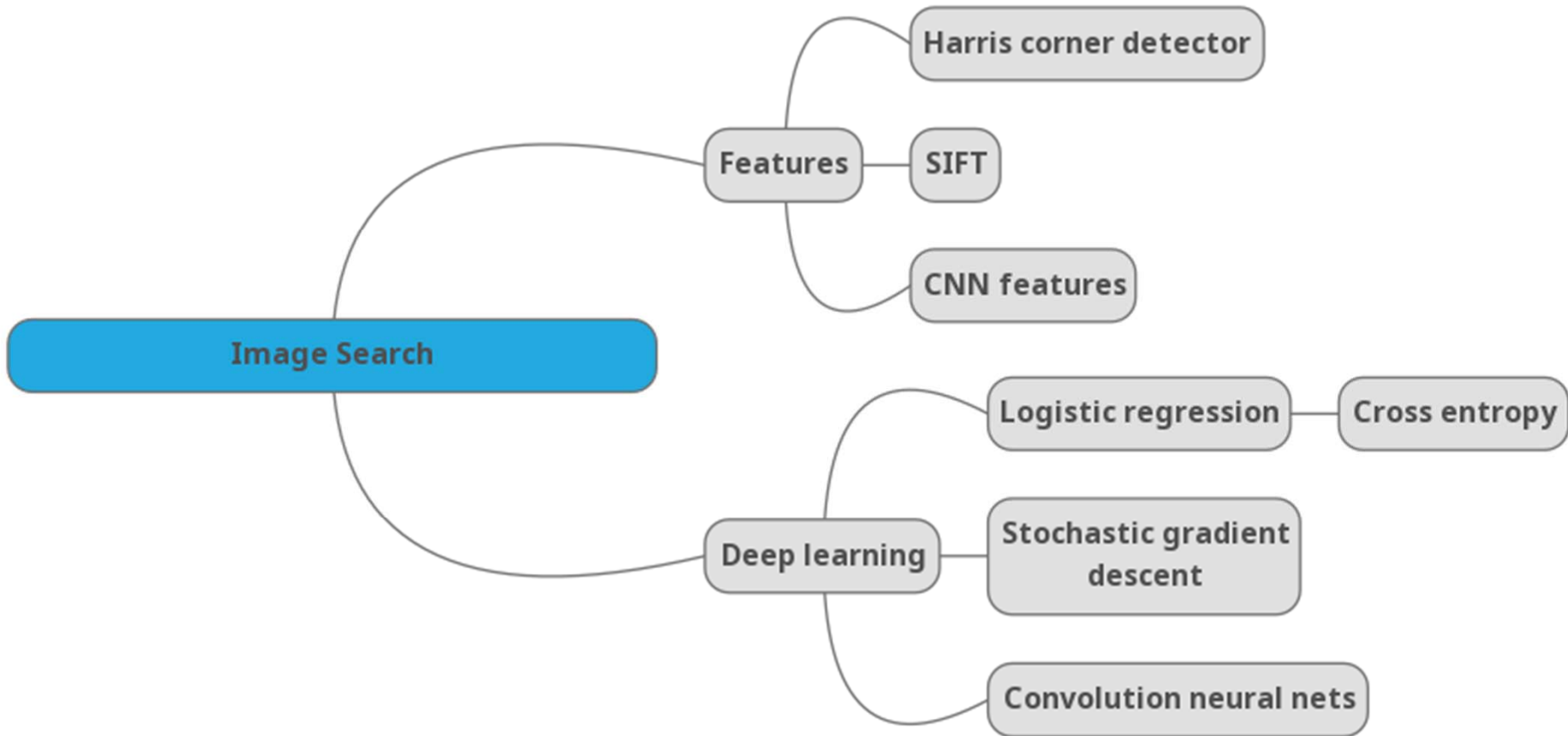
# Class Objectives were:

---

- **Study convolution neural nets (CNNs)**
- **At the prior class:**
  - **Browsed main components of deep neural nets**

# Summary up to Now

---



# Next Time

---

- **Bag-of-visual-Words (BoW) model**

# Homework for Every Class

---

- **Go over the next lecture slides**
- **Come up with one question on what we have discussed today**
  - 1 for typical questions (that were answered in the class)
  - 2 for questions with thoughts or that surprised me
- **Write questions 3 times before the mid-term exam**
  - Write a question about one out of every four classes
  - Multiple questions in one time will be counted as one time
- **Common questions are compiled at [the Q&A file](#)**
  - Some of questions will be discussed in the class
- **If you want to know the answer of your question, ask me or TA [on person](#)**