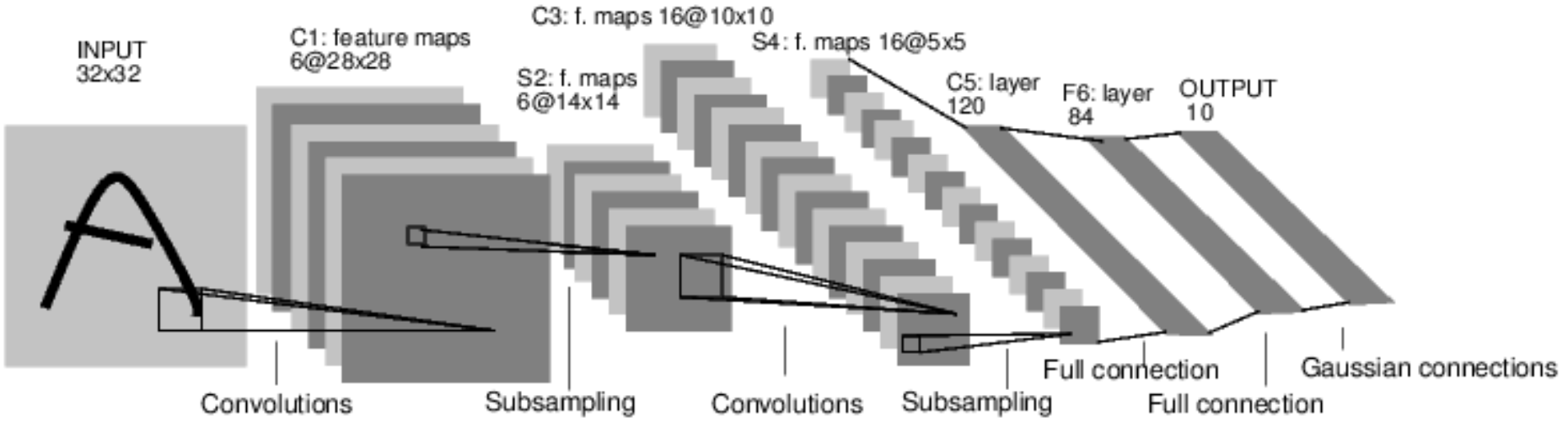
CSC 321 Assignment 2
Convolutional Networks

TA: Ryan Kiros
csc321ta@cs.toronto.edu



LeNet5 (LeCun et al 1998)

Motivation – ConvNets are everywhere!

(Krizhevsky et al, 2012)

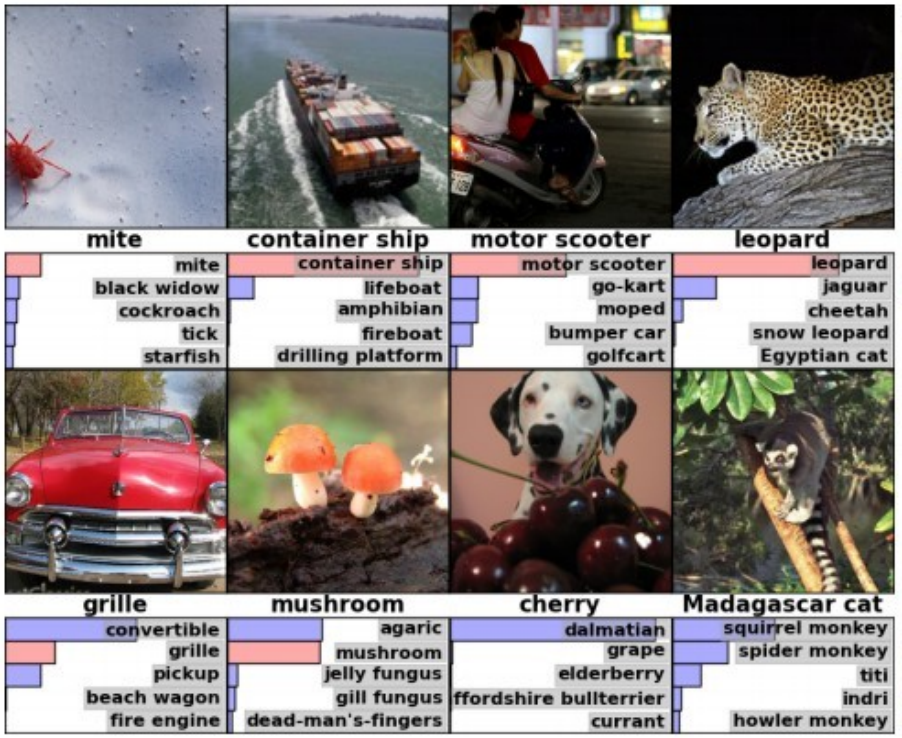
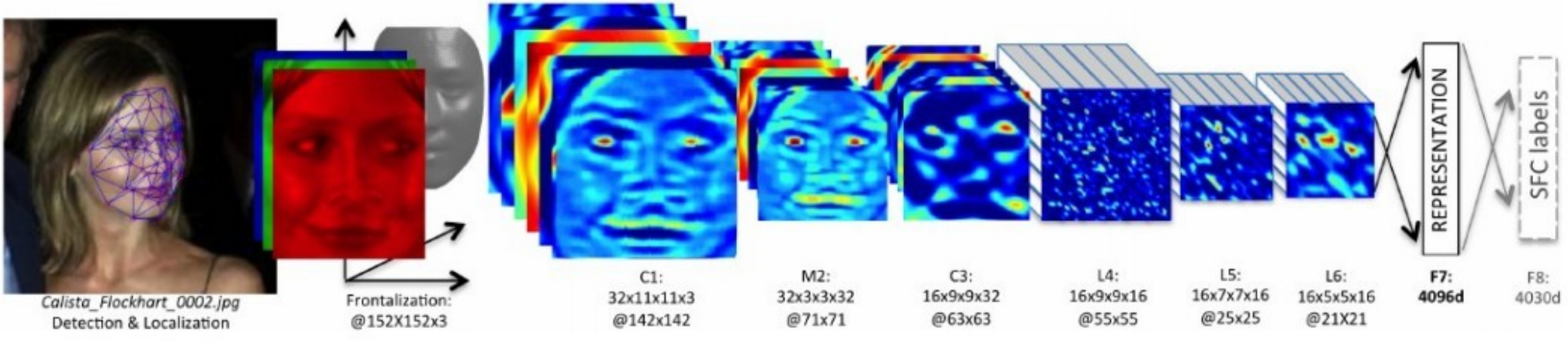


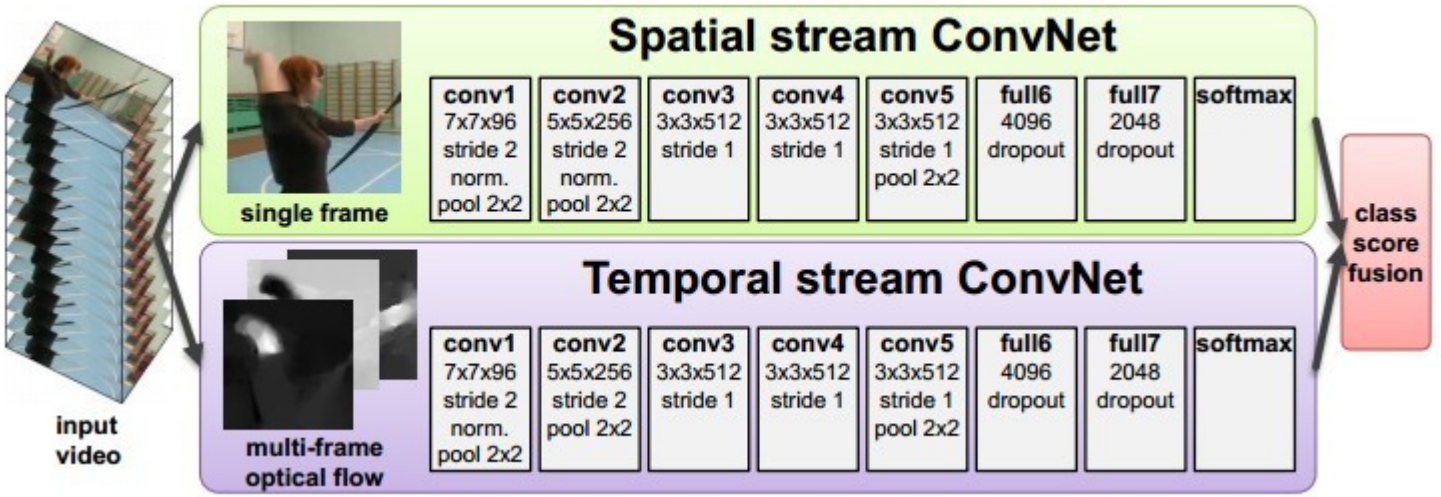
Image classification

Image retrieval

Motivation – ConvNets are everywhere!



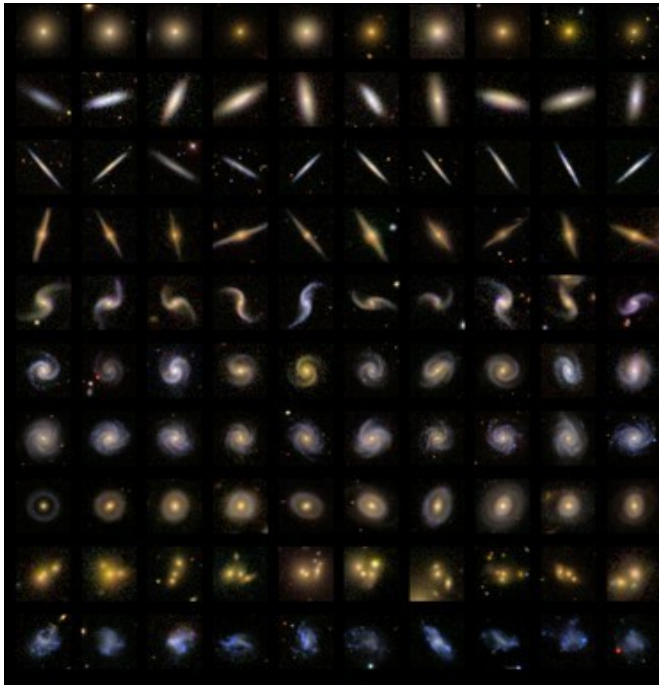
Face recognition (Taigman et al, 2014)



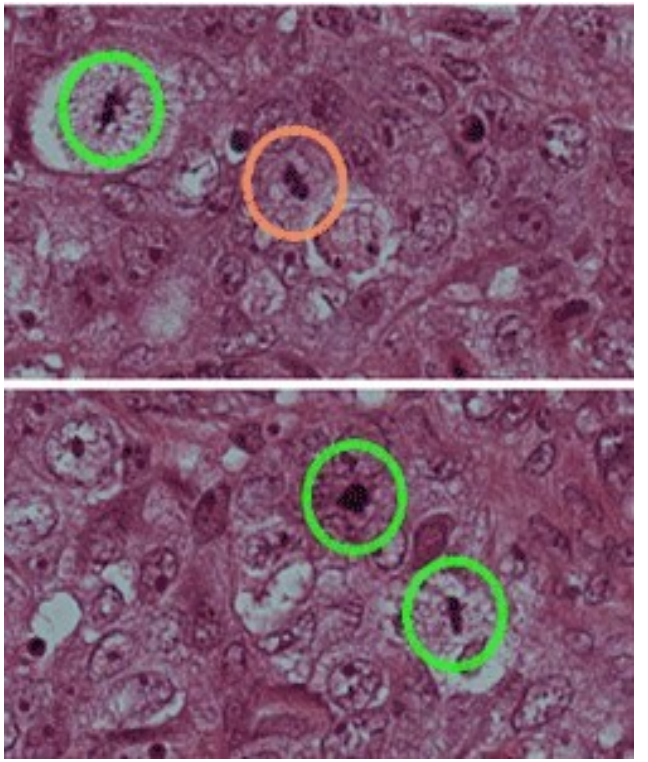
Action recognition from video (Simonyan et al, 2014)

Motivation – ConvNets are everywhere!

Street sign recognition
(Sermanet et al, 2011)



Galaxy classification
(Dieleman et al, 2014)



Mitosis detection
(Ciresan et al, 2013)

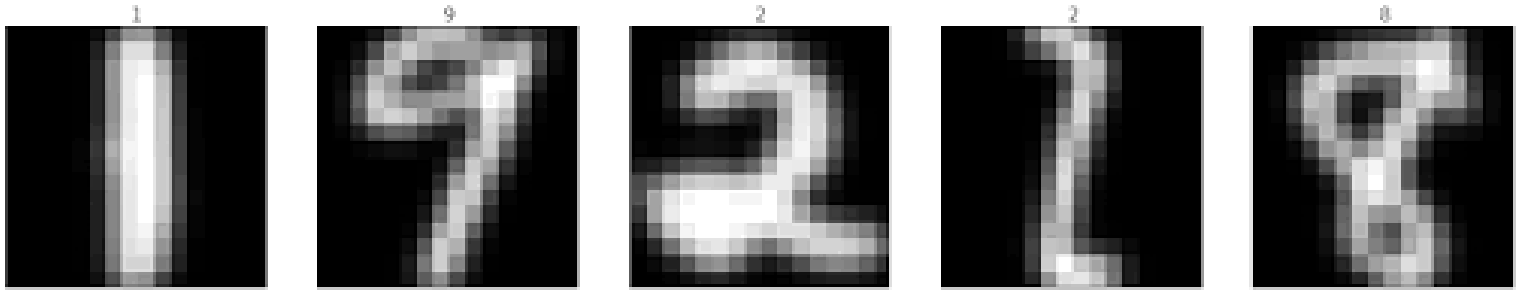
Motivation – ConvNets are everywhere!

Playing Atari games (Mnih et al, 2013)

- Many, many more applications (and not only vision):
  - Object detection
  - Image segmentation
  - Pose estimation
  - Image captioning
  - Pedestrian detection
  - Semantic image search
  - Extractive summarization
  - Sentiment analysis of text

This assignment

- You will train both fully connected and convolutional networks on the USPS handwritten digit dataset



(16 x 16 images of the digits 0 through 9)

- Goal: have a neural net that can accurately classify images of digits that it has never seen before

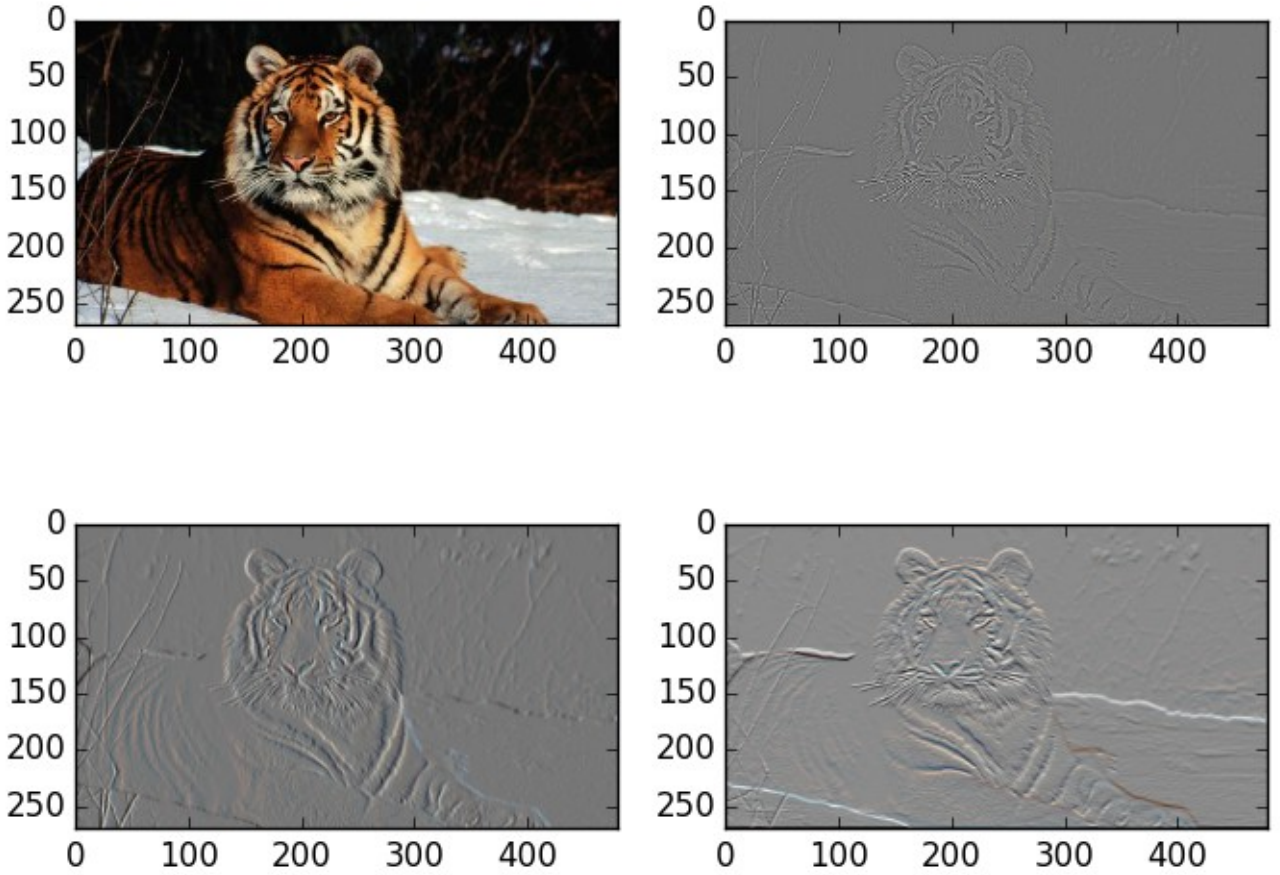
Assignment goals

- Once you have completed the assignment, you should have:
  - A better understanding of constructing neural nets In a modular way
  - Understand convolutions, and how they are used during the forward and backward passes in a convnet
  - Understanding of frequently used terminology (kernels, feature map, strides, padding, pooling, etc)
  - Enough understanding to learn more about all the cool examples from the beginning of these slides!

Assignment specifics

- Part I: Complete the forward and backward pass of a fully connected network
    - Just like in assignment 1, you only have to write a few lines of code: but each line requires you to think carefully.
    - Then train and analyze the results of a few models
- Part II: Complete the forward and backward pass of a convolutional network, using a given convolution function
    - This will be the most challenging part.
    - Finally, you will train and analyze several convnets, and answer questions about the model architectures

Playing with convolutions



- play_with_convolutions.py : gain intuition on the effect of convolving with different kernels

Specifying models: YAML

- Specify all the model details in a .yaml file

- The models used in this assignment are all preset. Though you are encouraged to try different settings yourself!

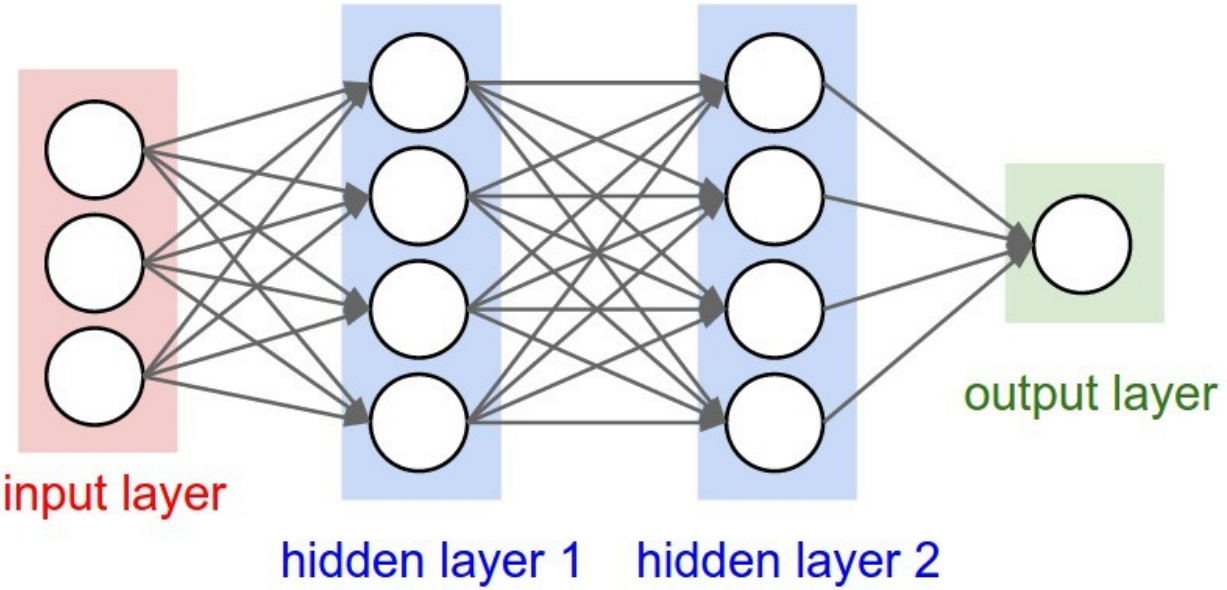
```
max_epochs: 25
batch_size: 100
data_path: usps.npz
checkpoint_file: net_1layer.npz
performance_stats: net_1layer_stats.npy
input_image_size_y: 16
input_image_size_x: 16
input_num_channels: 1
display: True
display_after: 20
network:
  - name : fc1
    type : FC
    num_channels : 512
    init_wt : 0.05
    epsilon: 0.02
    momentum: 0.9
    l2_decay: 0.0001

  - name : output
    type : FC
    num_channels : 10
    init_wt : 0.01
    epsilon: 0.02
    momentum: 0.9
    l2_decay: 0.0001
```

A brief review

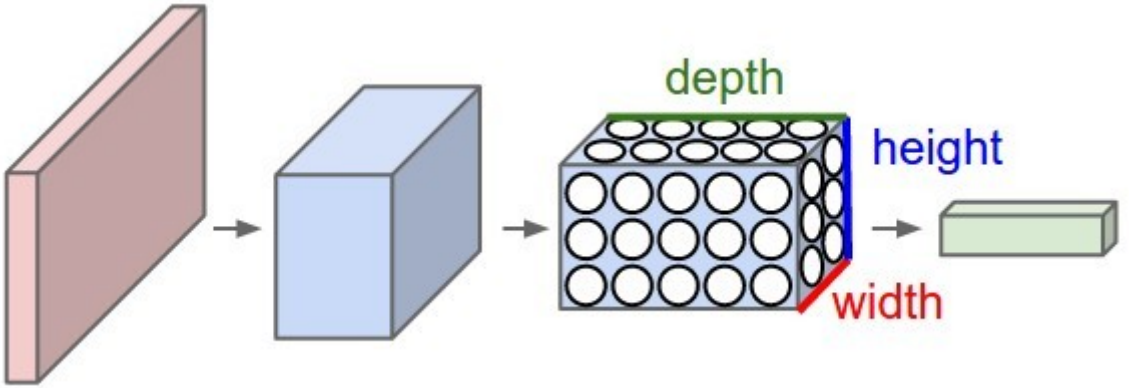
Fully connected:
(unique weights across all pairs of neurons)

Main operation:
Matrix Multiply

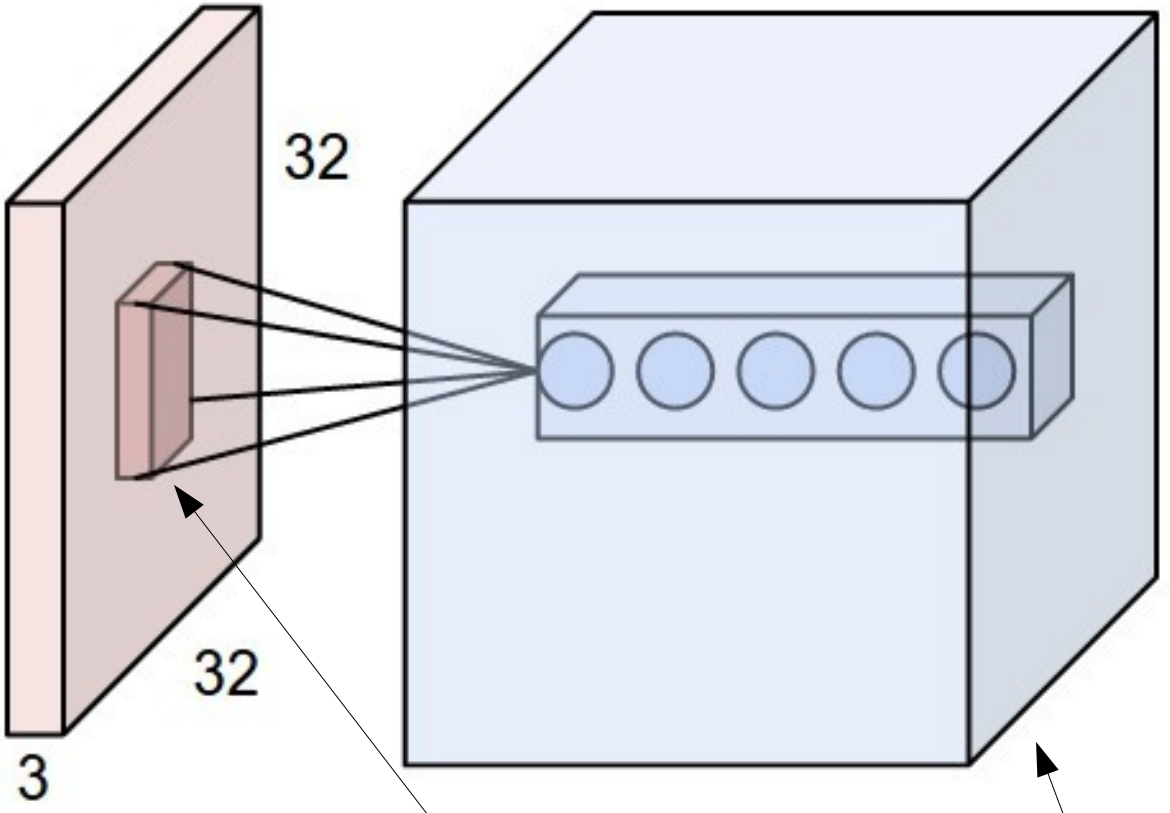


Convnet:
(neurons are volumes, weights are shared)

Main operation:
Convolution



Some terminology



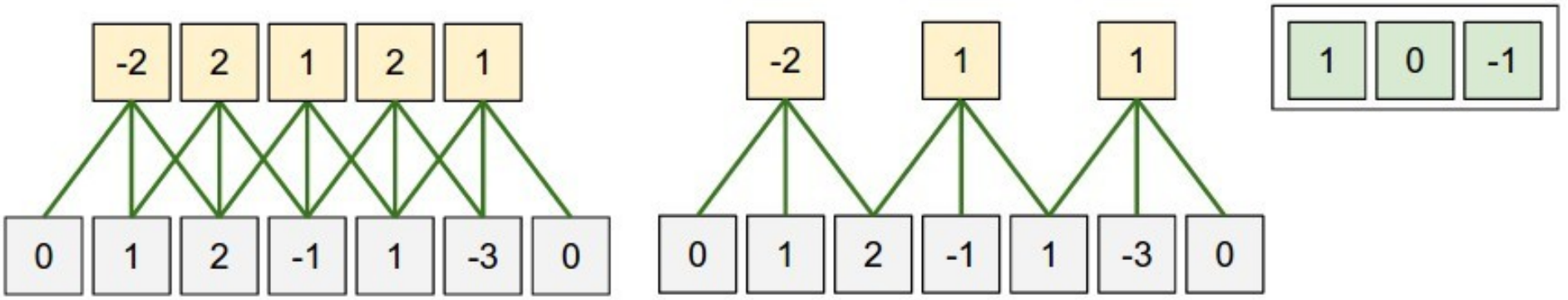
(think of this just like an image, but with 5 channels instead)

Channels
(e.g. 3 for RGB image)

Kernel (or filter)
5 in this example

Each “slice” across depth
Is a feature map

1D forward pass, strides, padding



Stride of 1

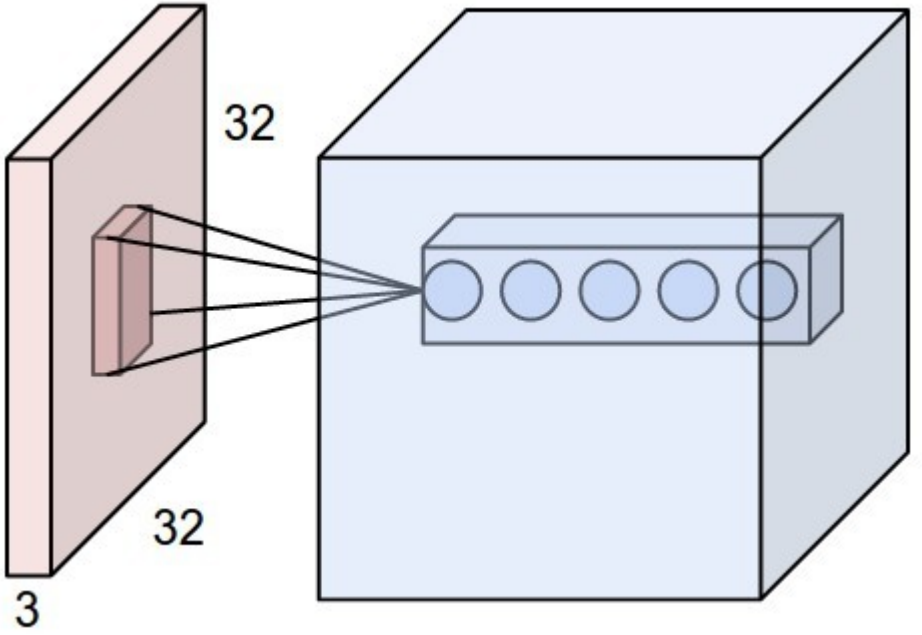
Stride of 2

kernel

- Weight sharing: the kernel is scanned across the input (as opposed to fully connected networks)
- Larger strides reduce computation cost, but usually at the expense of accuracy
- In this example, each side is “padded” with an extra 0

Example #1

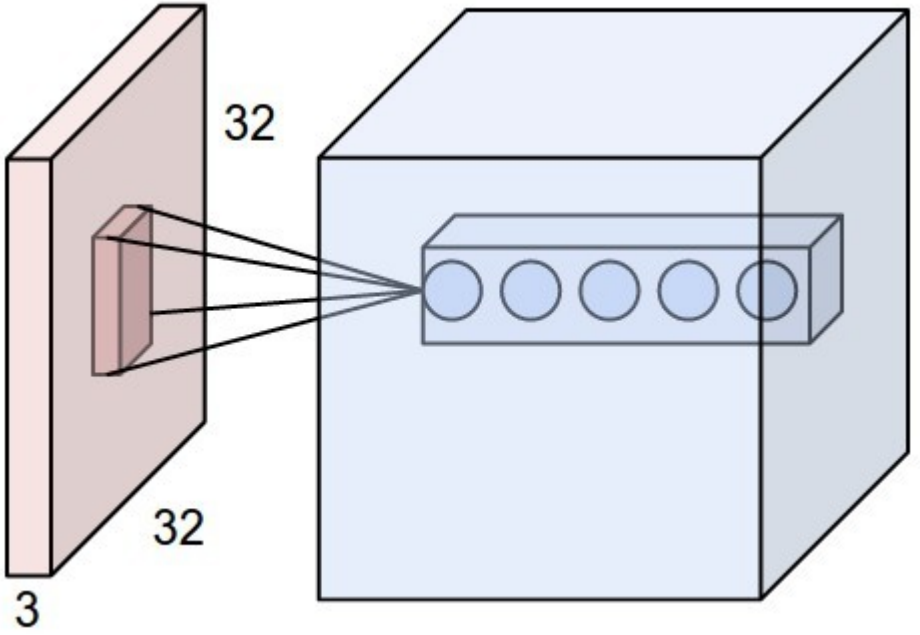
- Input: 32 x 32 x 3 image
- 5 Filters, each 5 x 5
- Stride of 1
- No padding



- What is the output volume?
- How many parameters are there?

Example #1

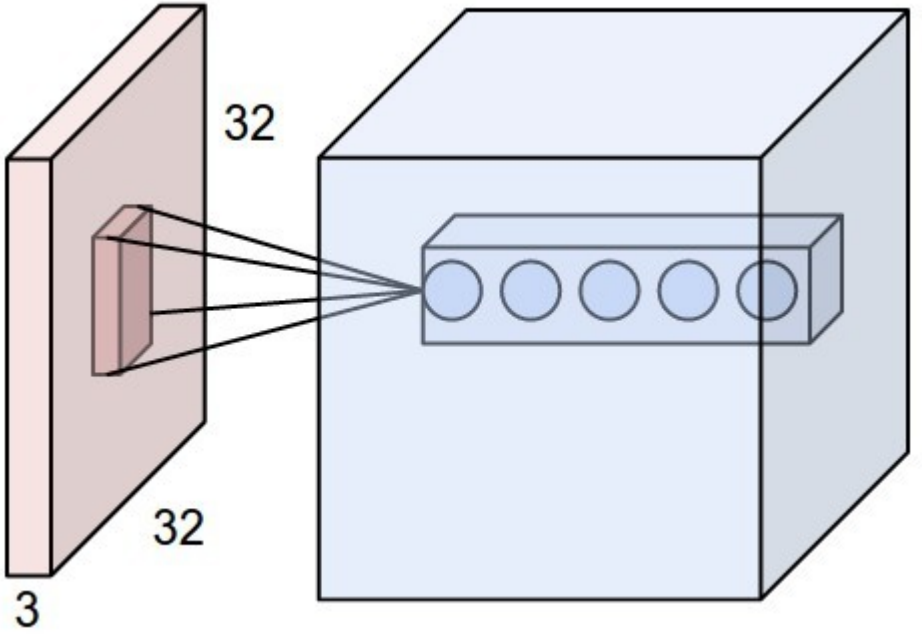
- Input: 32 x 32 x 3 image
- 5 Filters, each 5 x 5
- Stride of 1
- No padding



- What is the output volume? 28 x 28 x 5
- How many parameters are there? ((5 x 5) x 3) x 5 = 375

Example #2

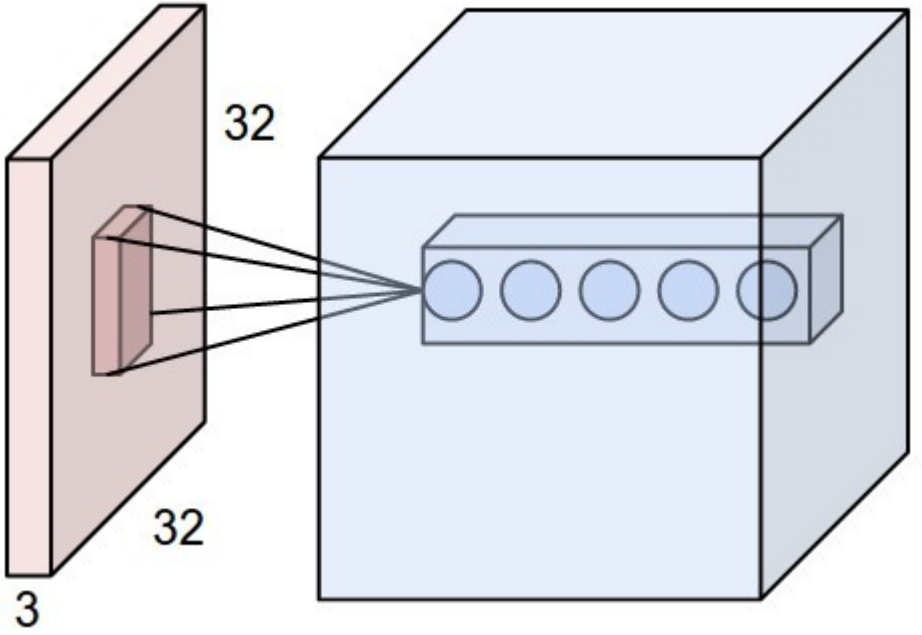
- Input: 32 x 32 x 3 image
- 5 Filters, each 5 x 5
- Stride of 3
- No padding



- What is the output volume?
- How many parameters are there?

Example #2

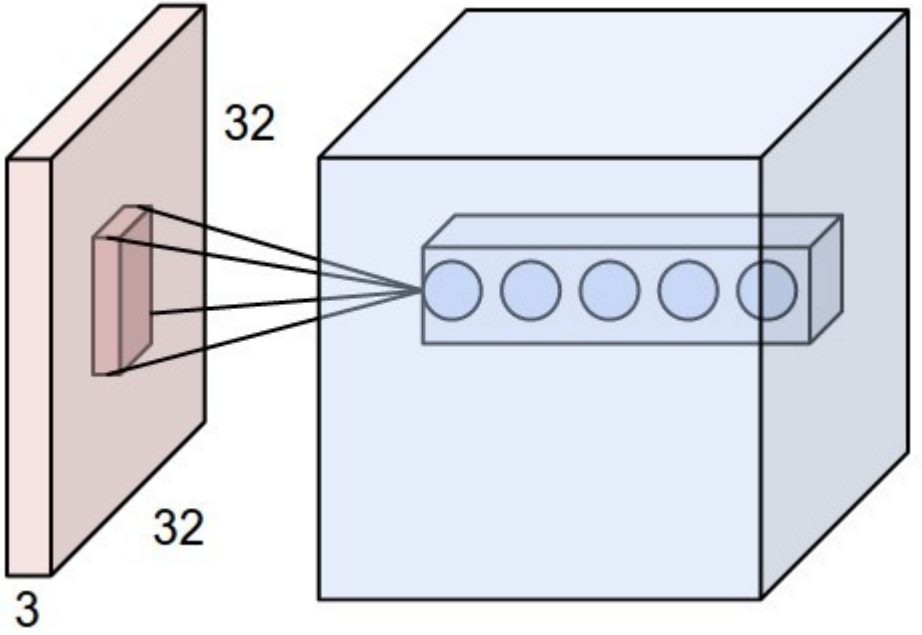
- Input: 32 x 32 x 3 image
- 5 Filters, each 5 x 5
- Stride of 3
- No padding



- What is the output volume? 10 x 10 x 5
- How many parameters are there? ((5 x 5) x 3) x 5 = 375

Example #3

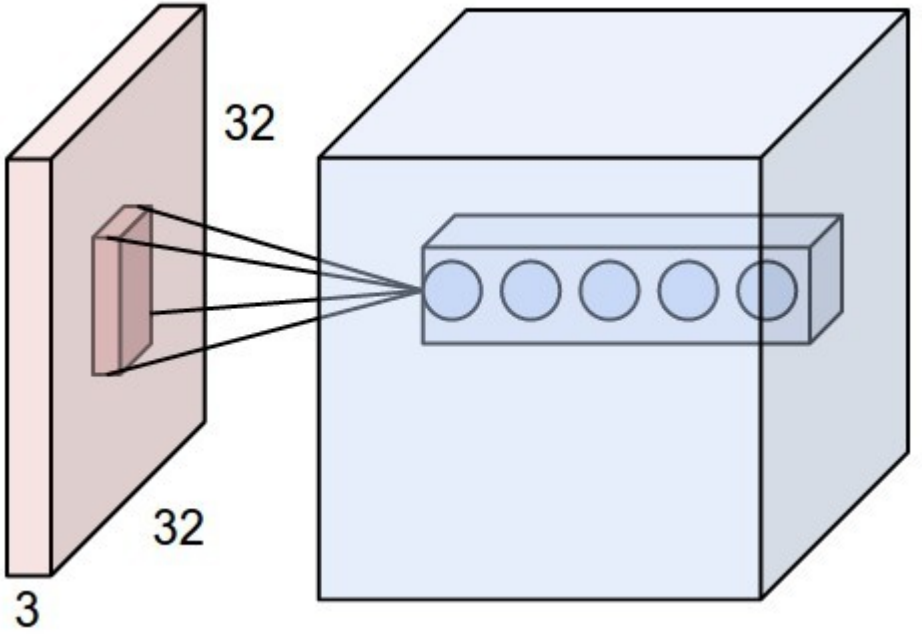
- Input: 32 x 32 x 3 image
- 5 Filters, each 3 x 3
- Stride of 1
- Padding of 1



- What is the output volume?
- How many parameters are there?

Example #3

- Input: 32 x 32 x 3 image
- 5 Filters, each 3 x 3
- Stride of 1
- Padding of 1



- What is the output volume? 32 x 32 x 5
- How many parameters are there? ((3 x 3) x 3) x 5 = 135

Additional resources

- Metacademy
- CS321n: Convolutional neural networks (Stanford)
  (Where I stole all the figures from)
- Stanford UFLDL tutorial on convnets (Matlab)
- deeplearning.net tutorial on convnets (Theano)

Deadline: Tuesday March 10, 2015 (at the start of class)