Deep Learning for Images

CMSC 389A: Lecture 7

Sujith Vishwajith

University of Maryland

Agenda

1. Applications
2. Convolutions
3. Convolution Layer
4. Convolutional Network
5. Announcements

Applications

Applications

Deep learning for image based tasks are used everywhere!

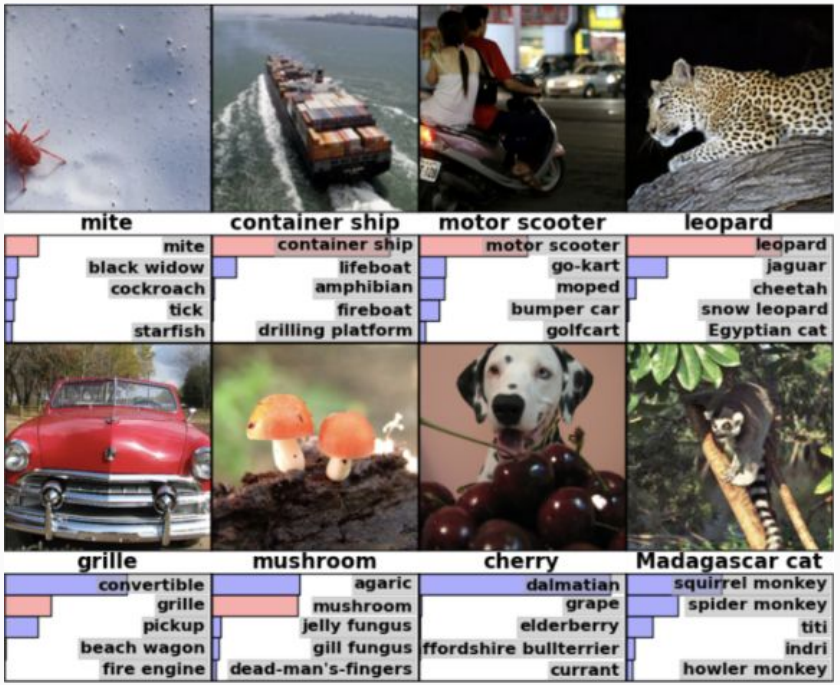
Due to recent improvement in GPUs.

Taking advantage of increase in image based data available..

Allows for automatic feature extraction compared to older methods (e.g. SVM).

Applications (cont.)

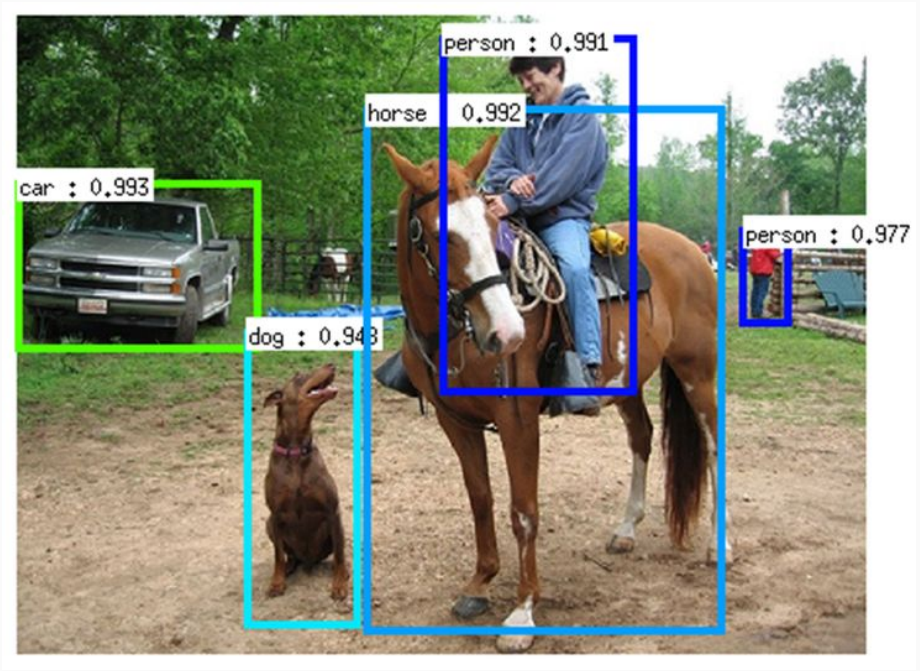
1. **Classification**
2. Object Detection
3. Segmentation
4. Retrieval
5. Art



ImageNet Classification Challenge

Applications (cont.)

1. Classification
2. **Object Detection**
3. Segmentation
4. Retrieval
5. Art



R-CNN used to detect and classify objects in images.

Applications (cont.)

1. Classification
2. Object Detection
3. **Segmentation**
4. Retrieval
5. Art

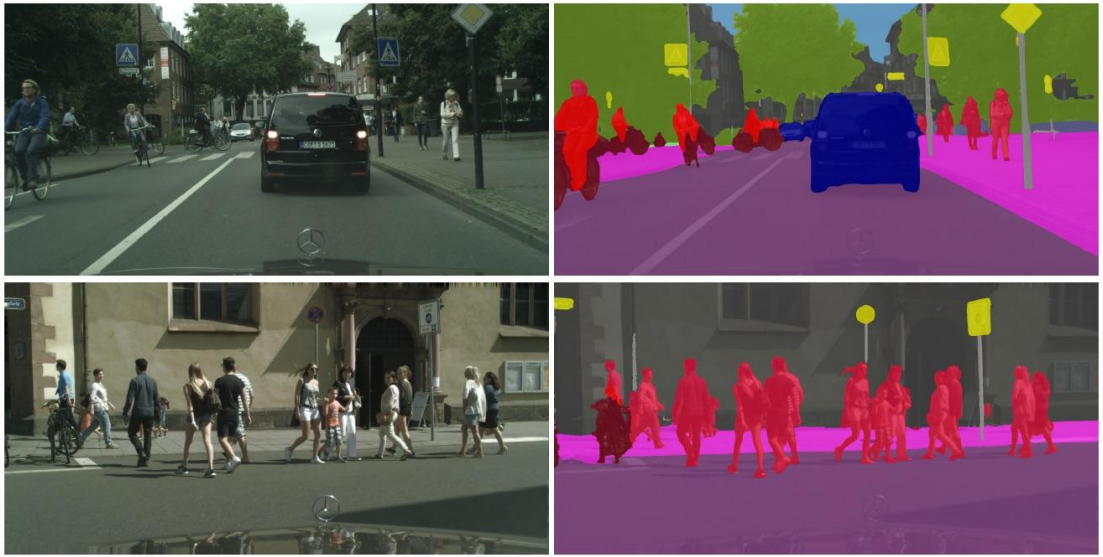
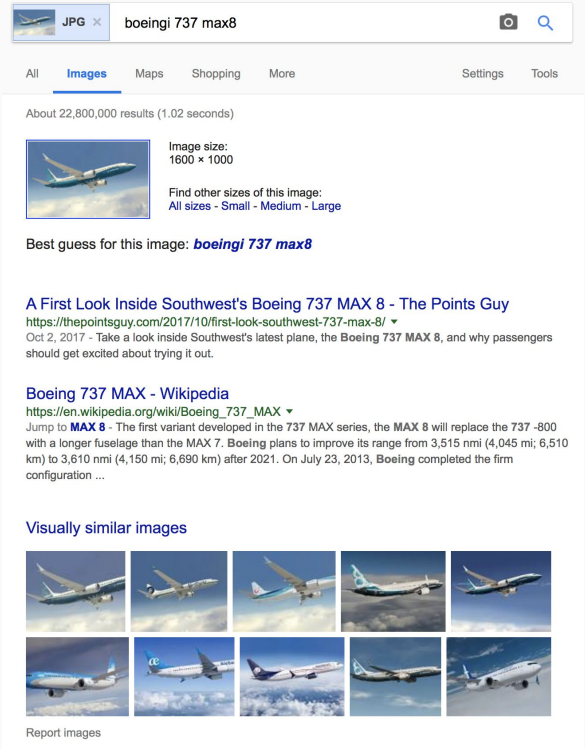


Image Segmentation for Self-Driving Cars.

Applications (cont.)

1. Classification
2. Object Detection
3. Segmentation
4. **Retrieval**
5. Art



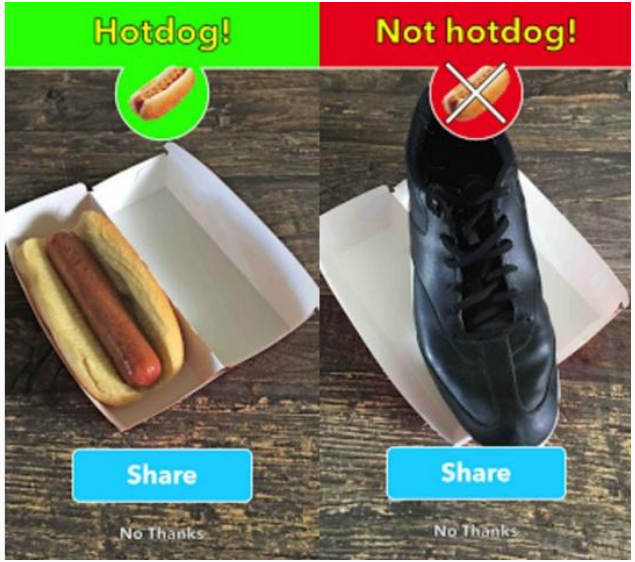
Search using images on Google.

Applications (cont.)

1. Classification
2. Object Detection
3. Segmentation
4. Retrieval
5. **Art**

Neural Style-Transfer for Imitating Painter Styles

Not Hotdog



The Not Hotdog app from HBO's Silicon Valley used CNN's.

Inspiration

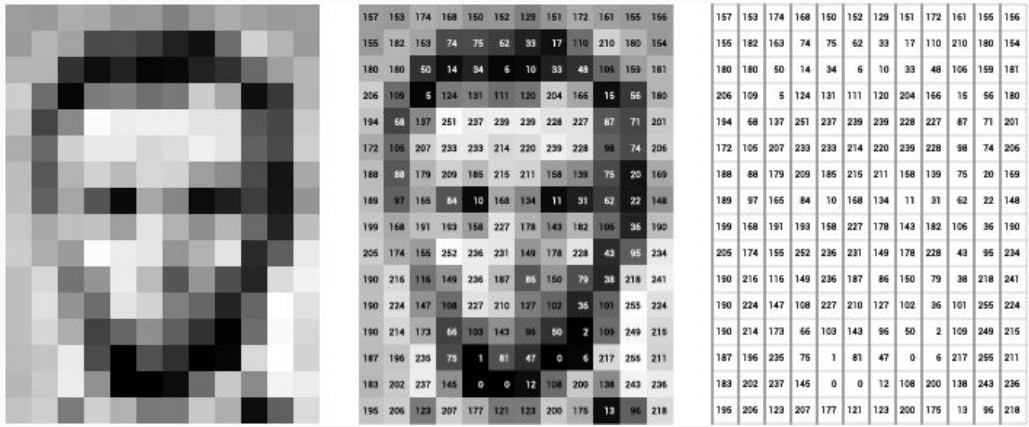
Image Representations

Images are represented as a matrix of pixels.

Dimensions of the matrix is also known as the “resolution” (H x W) of the image.

Grayscale: Pixel have 1 value between 0 (black) and 255 (white). *H x W x 1*

Color: Pixels have 3 values between 0 and 255. One for red, green, and blue. *H x W x 3*



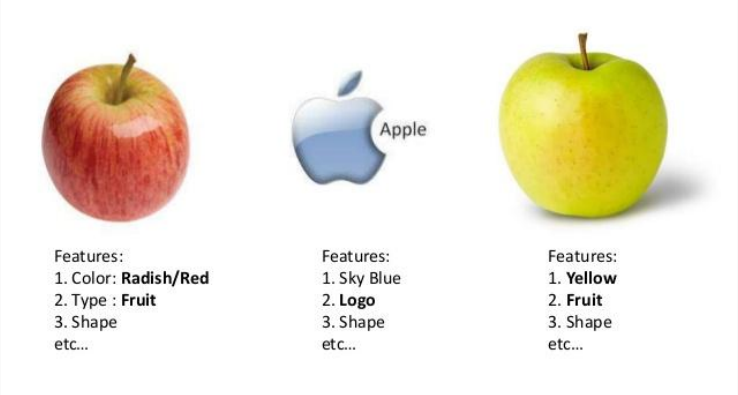
Inspiration

Features of an image tell us what is in the image.

Previous methods required us to design features manually (feature engineering).

Performance of a model is directly correlated with the quality of features.

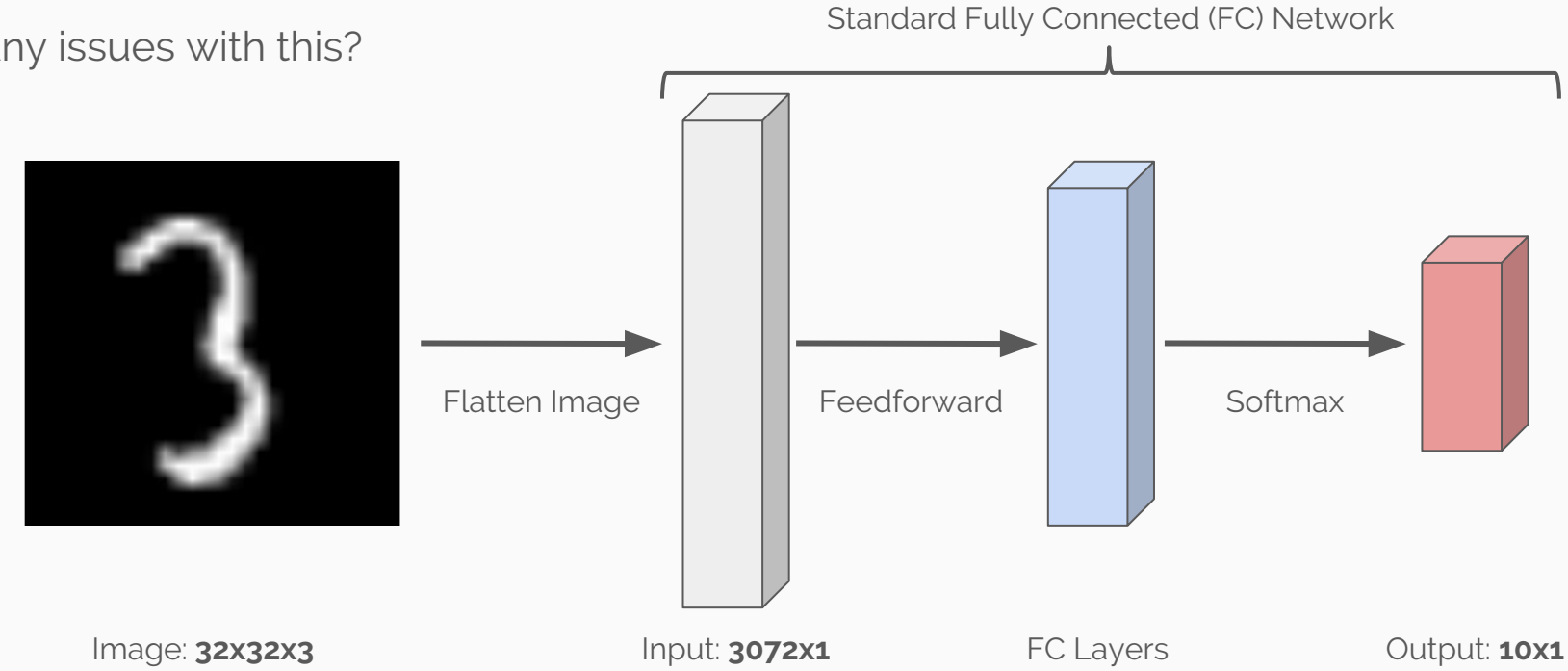
Could we automatically learn good features?



Inspiration (cont.)

Can each pixel be an input feature?

Any issues with this?



We never used any spatial information!

Fix: Convolutions

Convolutions

Convolutions

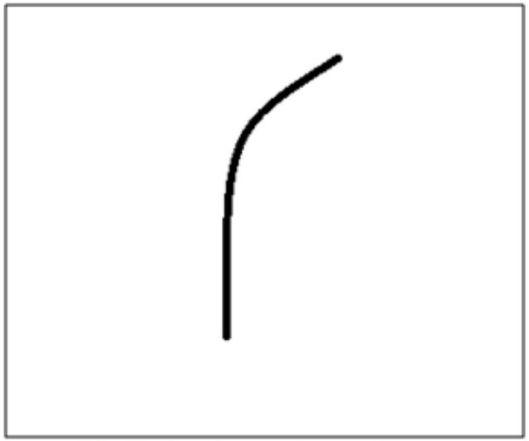
Filter of dimension HxWxD that “convolves” around an image

Each filter represents a certain feature (e.g. a line) and is a matrix of weights

If a region contains that feature, it will match highly with the filter

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter



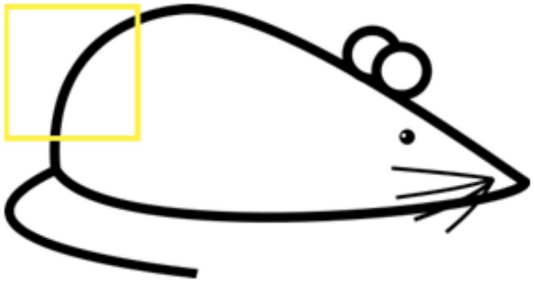
Visualization of a curve detector filter

Convolutions (Cont.)

We slide the filter through the image and compute its score at each region

Score is multiplication of value in filter with corresponding pixel value in region

Score is typically passed through an activation function (e.g ReLU)



Visualization of the filter on the image

Visualization of the receptive field

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

Pixel representation of the receptive field

*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = (50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600 (A large number!)

Convolutions Example

Applying a 3x3 filter to first region

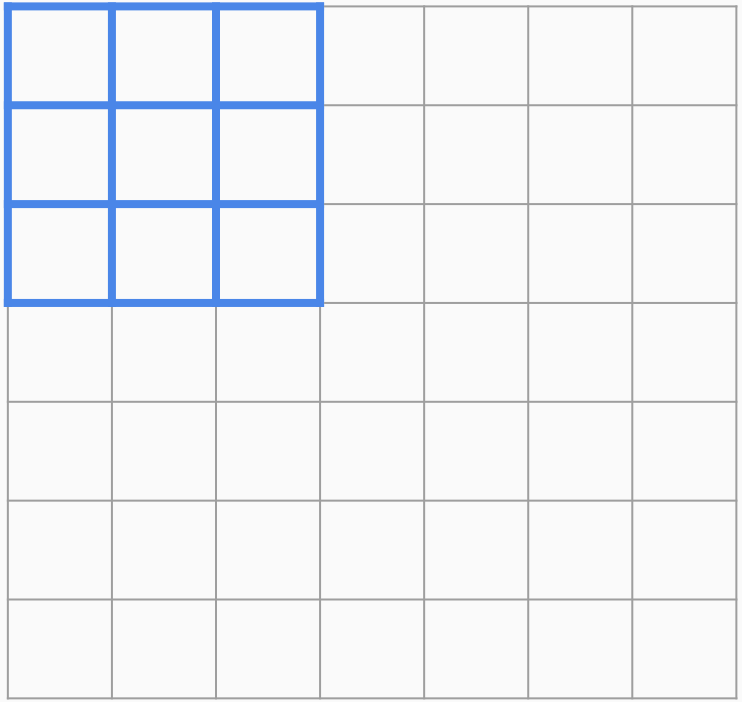


Image: **7x7x1**

Output: **5x5x1**

Convolutions Example (cont.)

Applying a 3x3 filter to second region

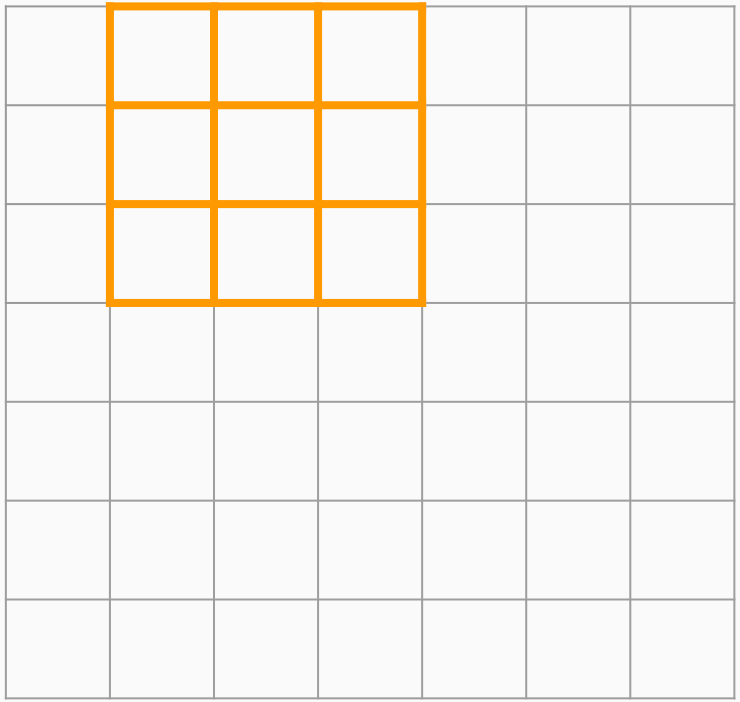
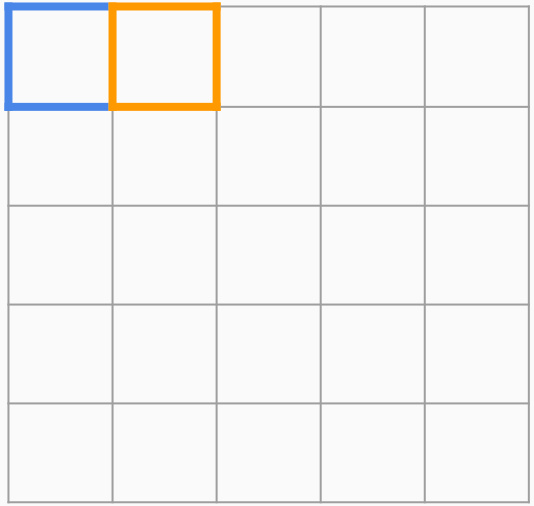


Image: **7x7x1**



Output: **5x5x1**

Convolutions Example (cont.)

Applying a 3x3 filter to sixth region

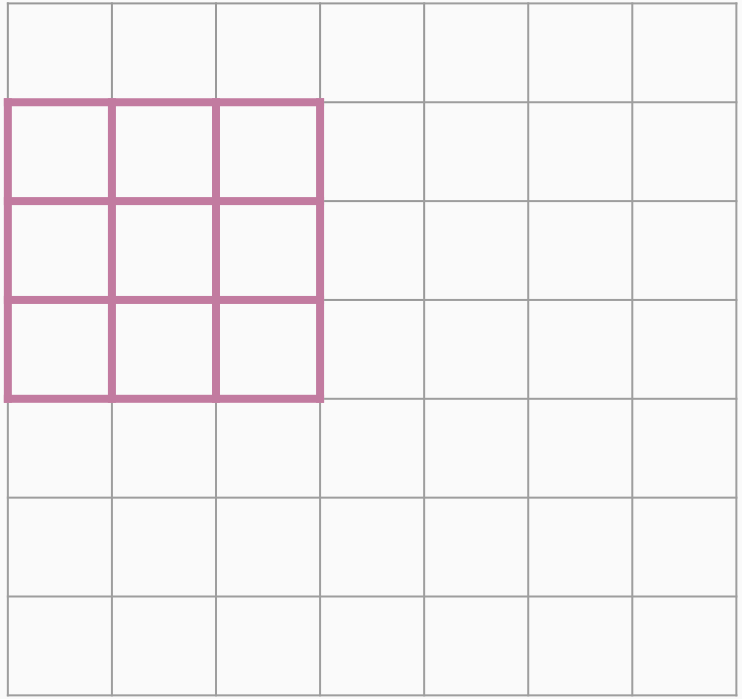
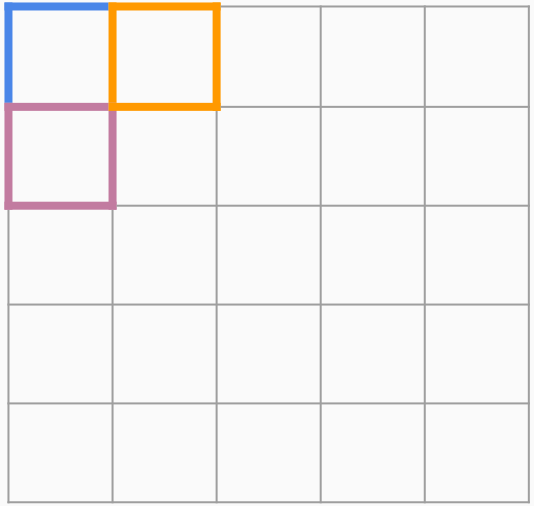
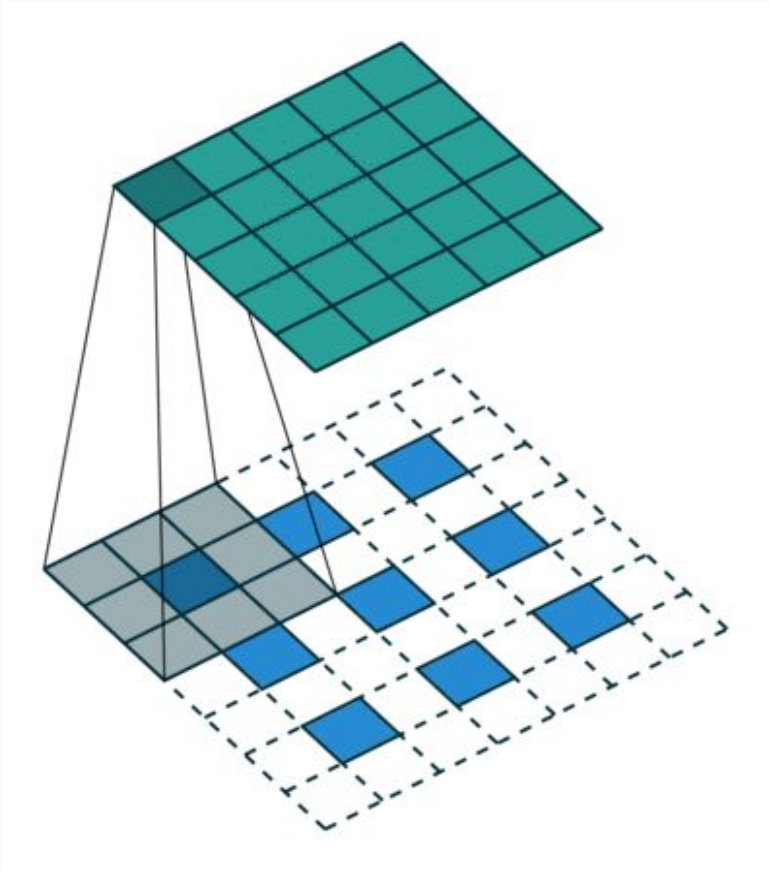


Image: **7x7x1**



Output: **5x5x1**

Visualization



Convolutions (cont.)

Repeat till we've covered every region in the image

Gives us an "activation map" for the image (how activated each region was for this filter)

We can change stride length which is how many pixels the filter jumps by

Notice how our output has a smaller dimension than our input. We can fix that by "zero" padding the input.

Why do we pad? Research shows that if the image decreases in size too quickly through the network, the model performs worse.

Increase Stride Length

What happens if we change the stride length to 2?

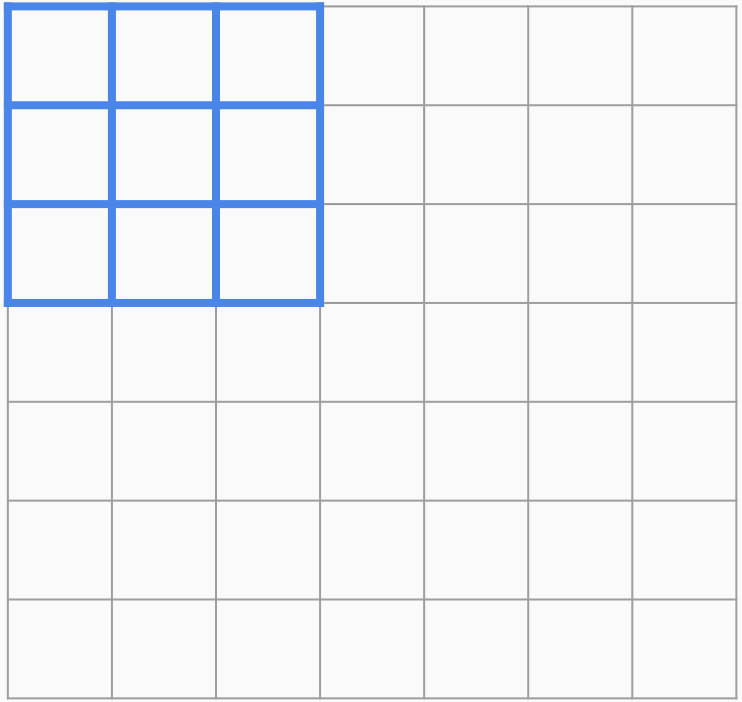
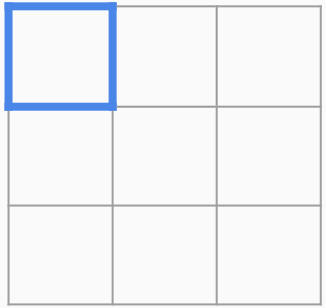


Image: **7x7x1**



Output: **3x3x1**

Increase Stride Length

What happens if we change the stride length to 2?

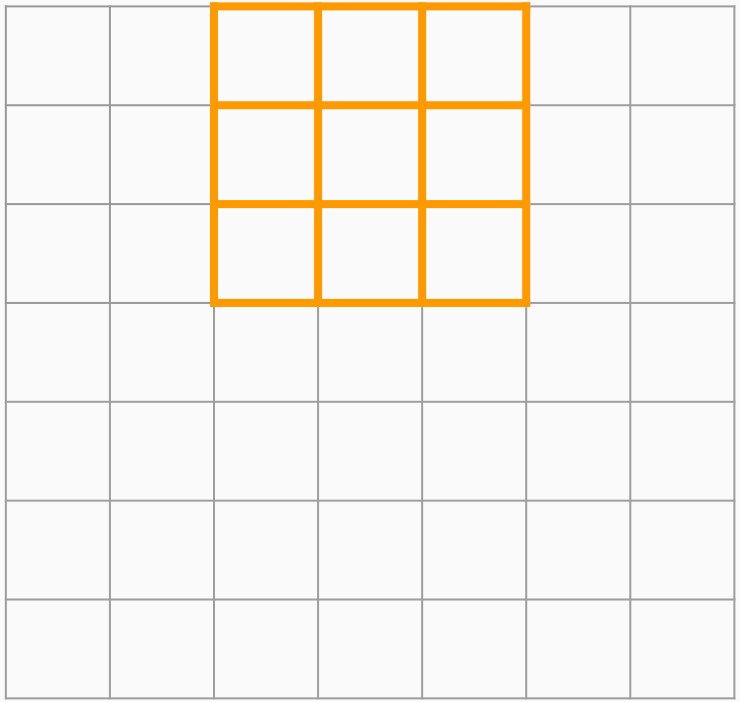
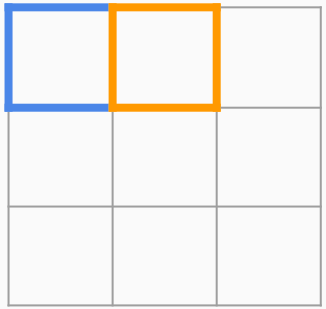


Image: **7x7x1**



Output: **3x3x1**

Increase Stride Length

What happens if we change the stride length to 2?

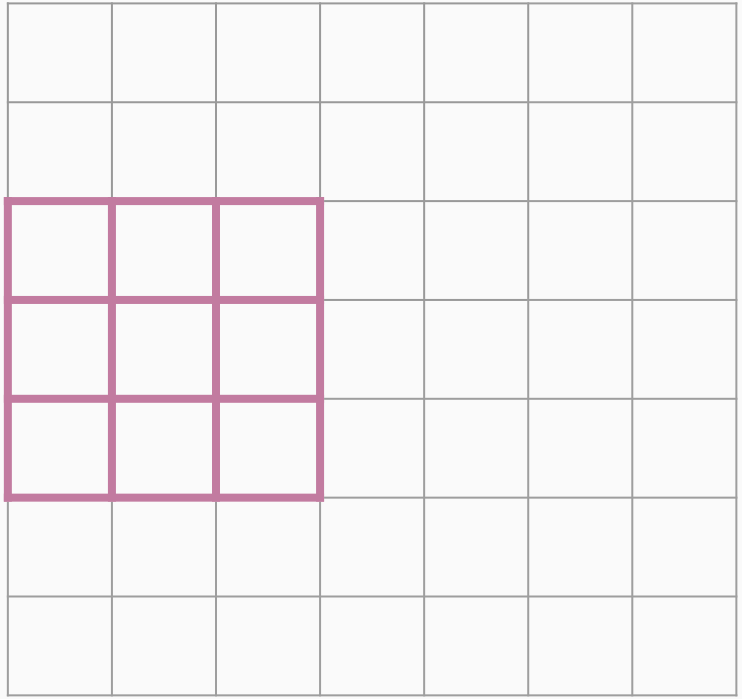
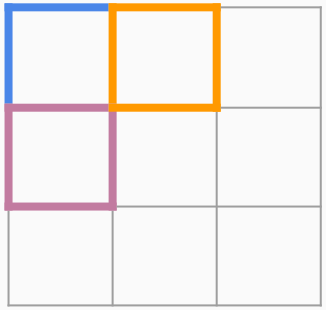


Image: **7x7x1**



Output: **3x3x1**

Zero Padding

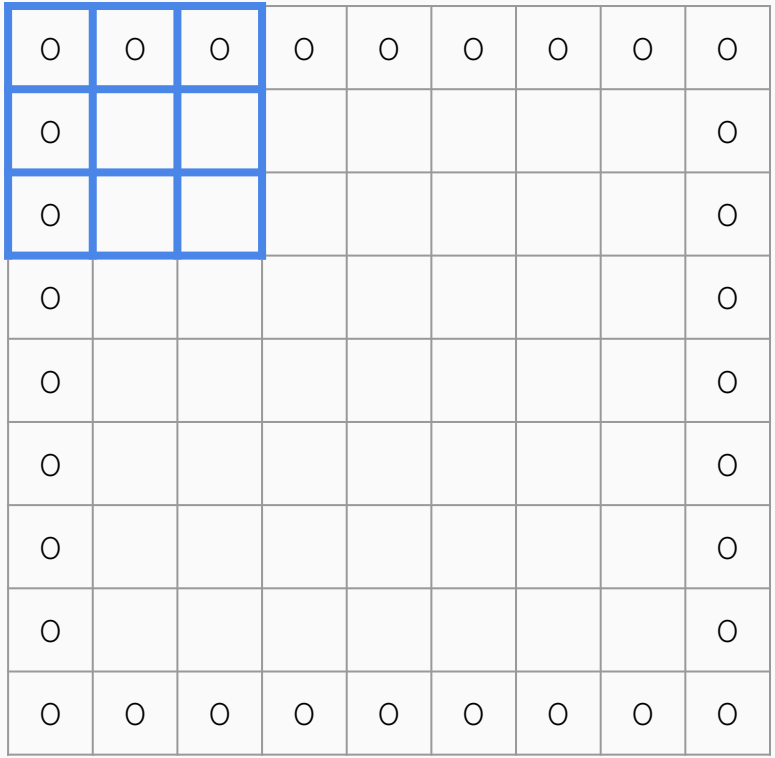
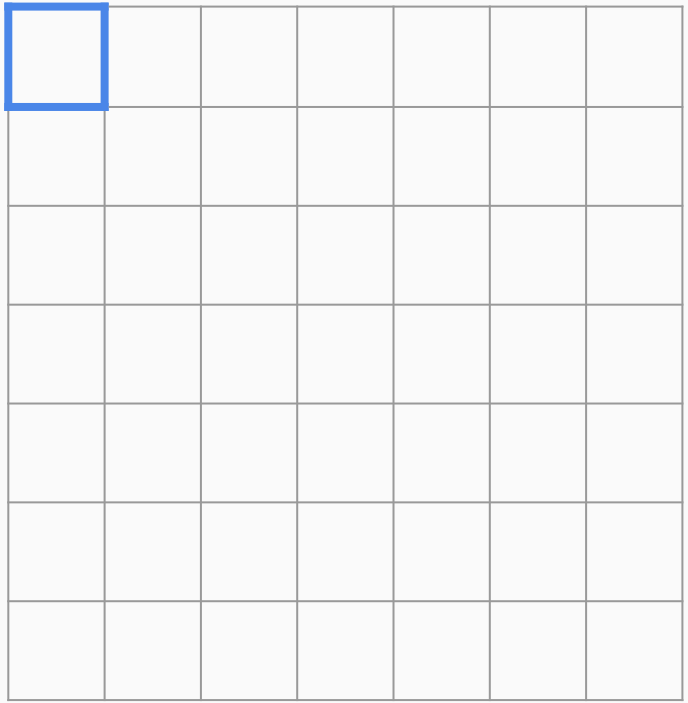


Image: **7x7x1** -> 1 Padded: **9x9x1**



Output: **7x7x1**

Convolutions (cont.)

Formula for calculating what your padding should be to keep same size

$$P = (K-1) / 2$$

K: filter size

Convolutions (cont.)

Formula for calculating output size:

$$O = ((W - K + 2*P) / S) + 1$$

O: output height/width

W: input height/width

K: filter size

P: padding amount

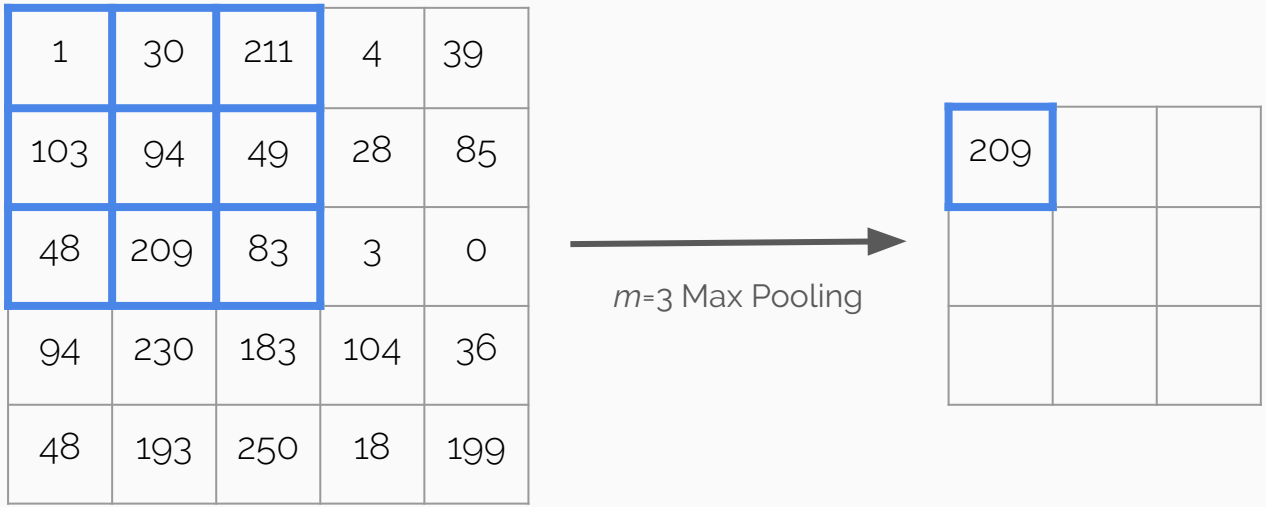
S: stride value

Pooling Layer

Used for downsampling (to reduce dimensions of image)

Most common is Max Pooling of filter size *m* which takes the max value in the region

Helps speed up computation and takes the “best” feature for a region



Let’s try an example.

Link to video containing example:
https://www.youtube.com/watch?v=2-Ol7ZB0MmU

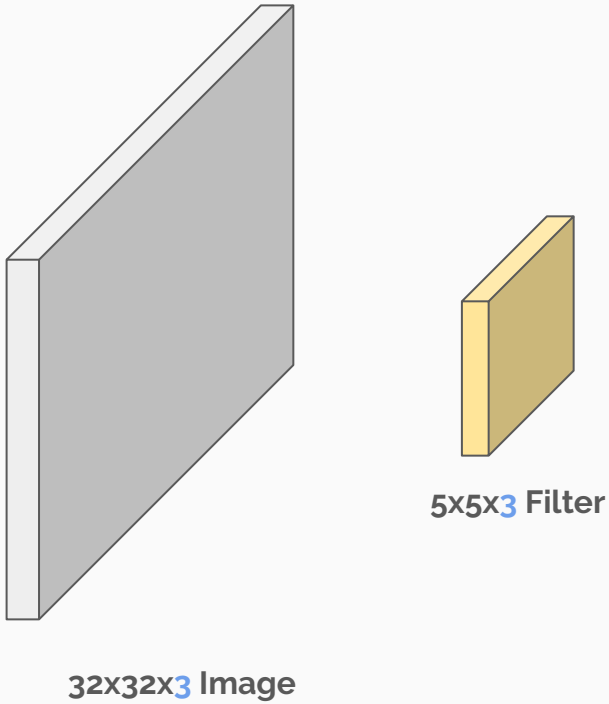
Convolutional Layer

Convolution Layer

Convolve filters sliding over the image

Notice filters always have same depth as image

Preserves spatial structure



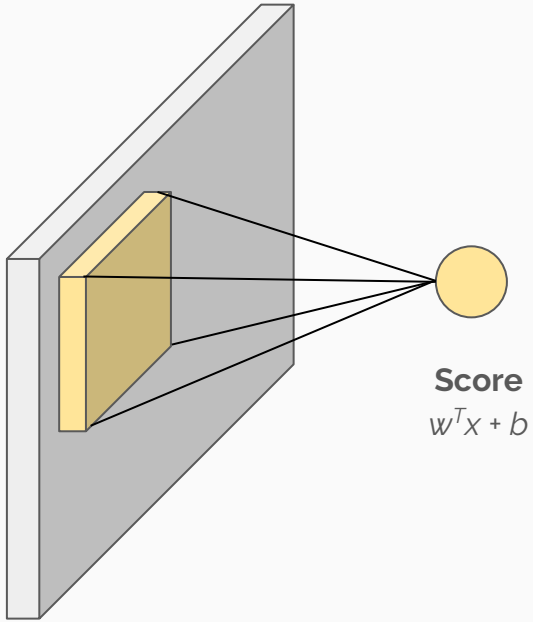
Convolution Layer (cont.)

Each filter serves as weights for it's feature

Take dot product of pixels in that region and filter

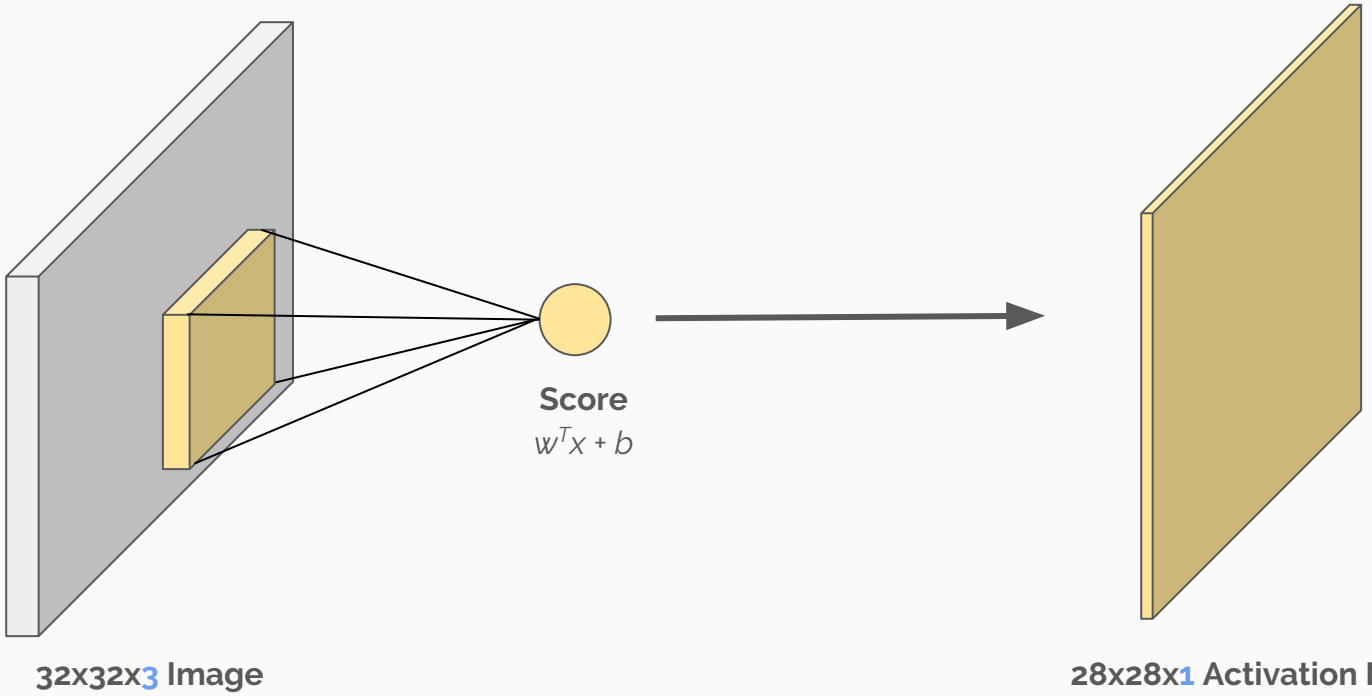
Pass it through an activation function (e.g. ReLU)

Gives us a "score" for that region



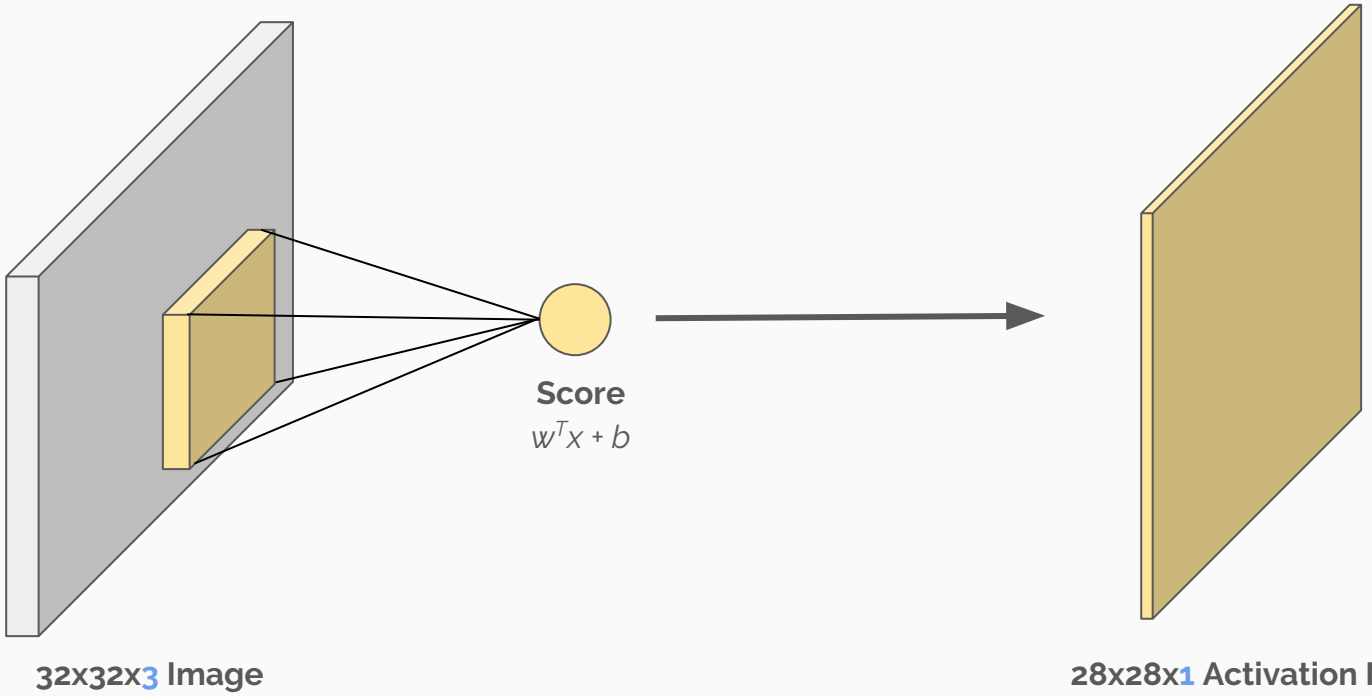
Convolution Layer (cont.)

We sliding the filter over the entire image to get an activation map for that feature



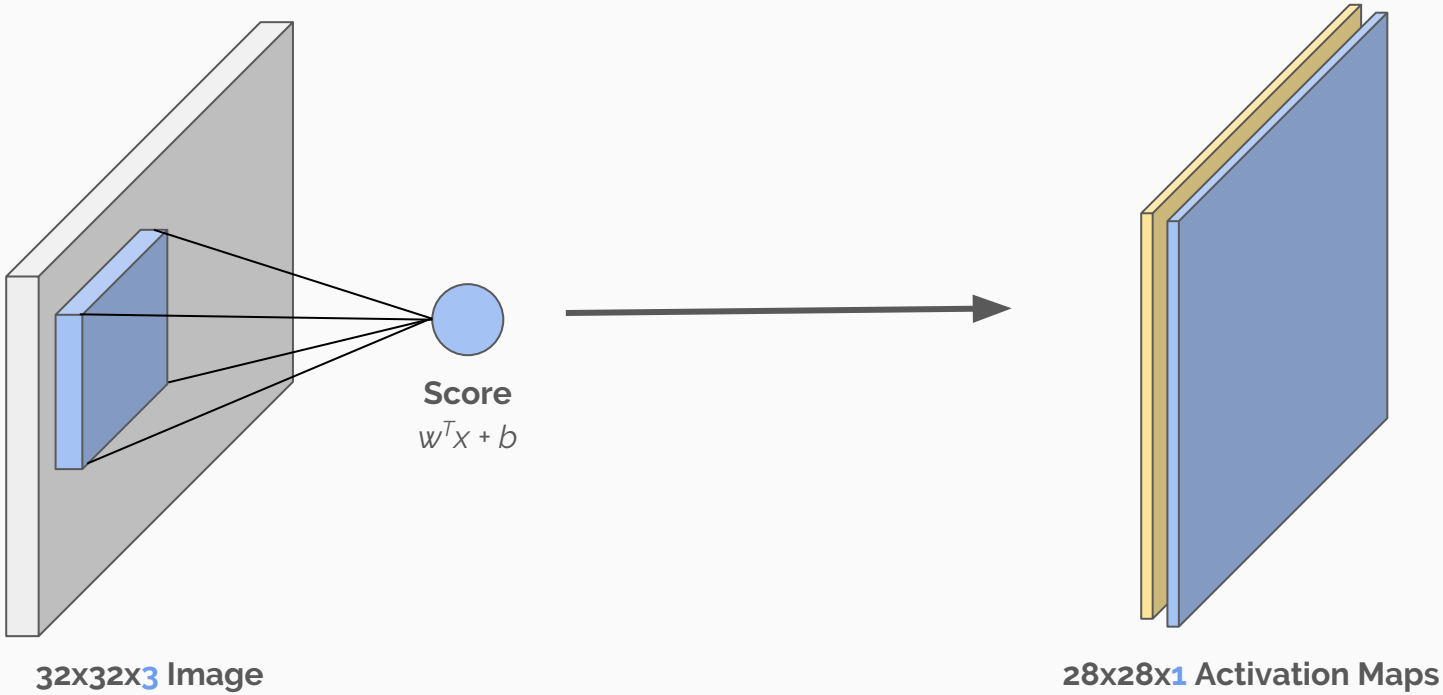
Convolution Layer (cont.)

We slide the filter over the entire image to get an activation map for that feature



Convolution Layer (cont.)

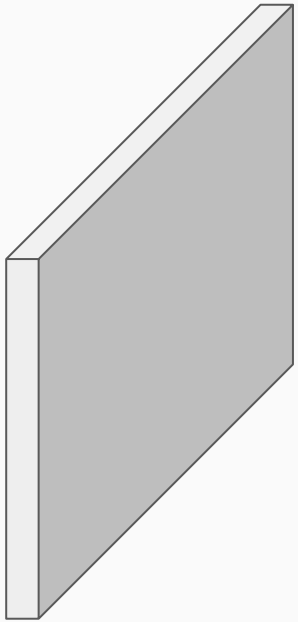
We can add another filter to get a second activation map



Convolution Layer (cont.)

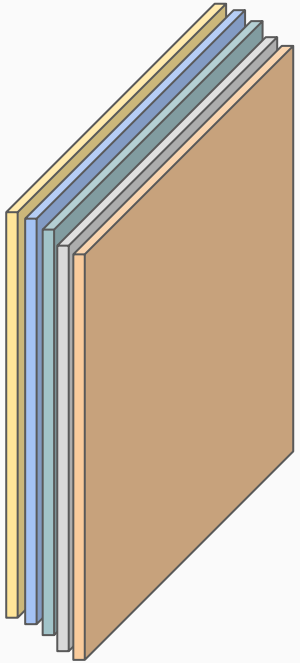
Repeat the process with five filters (each looking for a different feature)

Get a new image (of features) with dimension **28x28x5**



32x32x3 Image

Convolutions with 5 Filters



5 Activation Maps each 28x28x1

Parameters

Four Hyper-parameters:

1. Number of filters **K**
2. Filter size **F**
3. Stride **S**
4. Amount of padding **P**

Common Settings (from Andrej Karpathy):

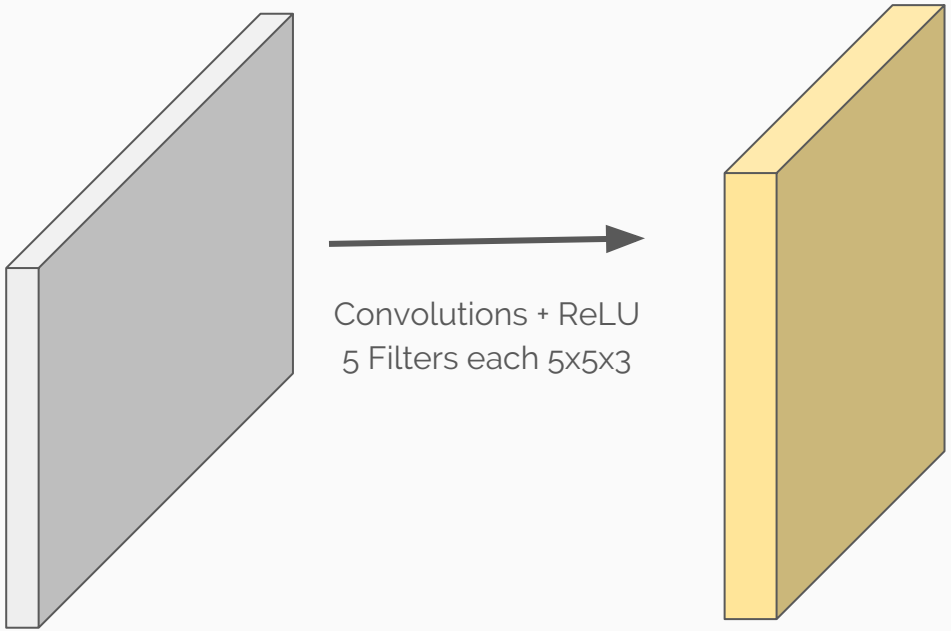
- K = powers of 2 (e.g. 32, 64, 128, 512)
- F = 3, S = 1, P=1
- F = 5, S = 1, P = 2
- F = 1, S = 1, P = 0

Convolutional Network

Convolutional Network

Pass the new image through another convolutional layer

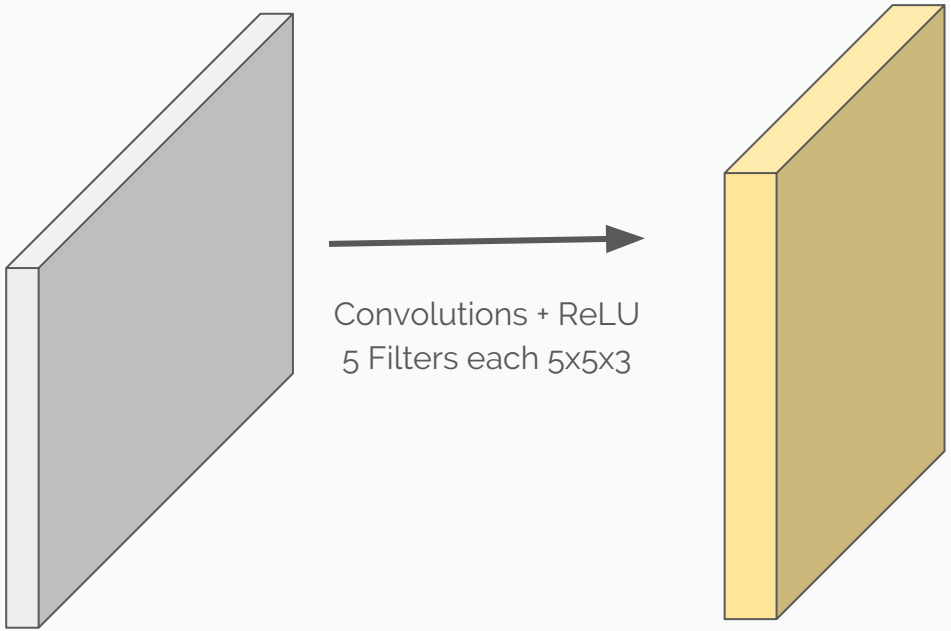
Goal is to get find higher level features from the old features



Convolutional Network (cont.)

Number of parameters?

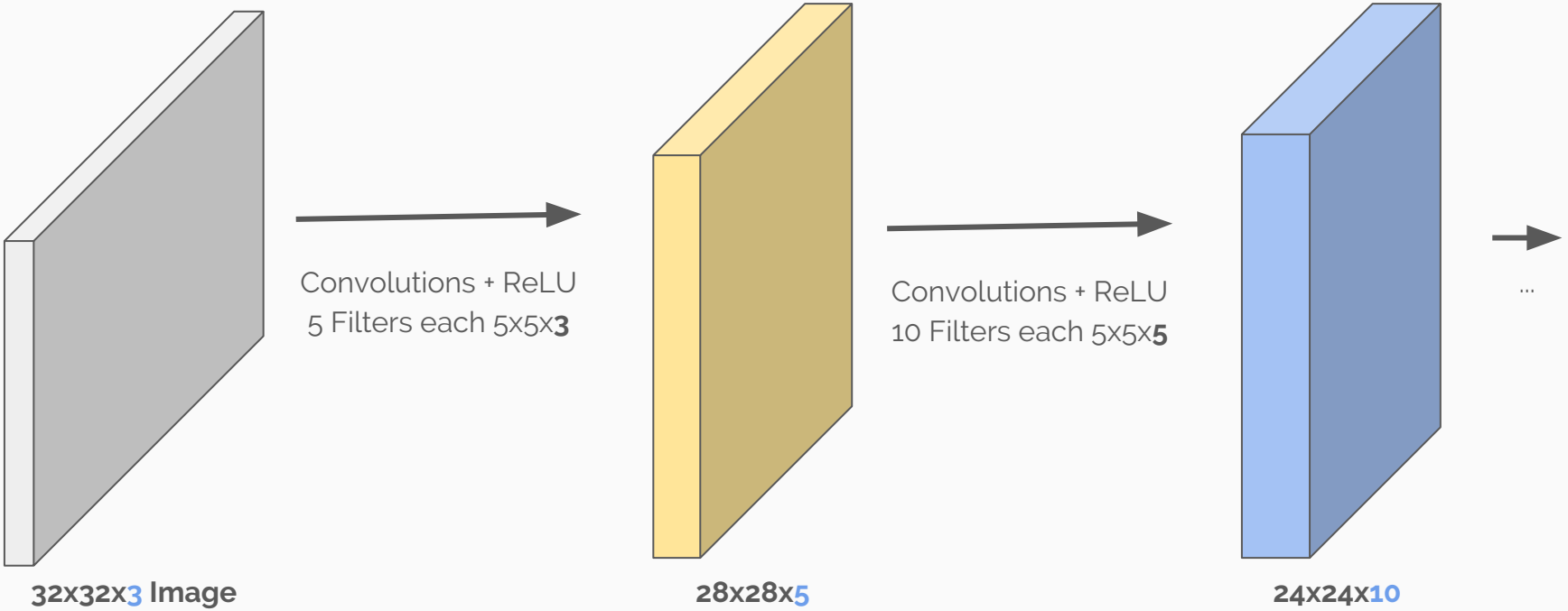
Each filter has 5*5*3=76 parameters and 5 filters so total of 380 parameters.



Convolutional Network (cont.)

A series of convolutional layers will help us find higher level features

E.g. Finding shapes from lines and then objects from shapes



Convolutional Network (cont.)

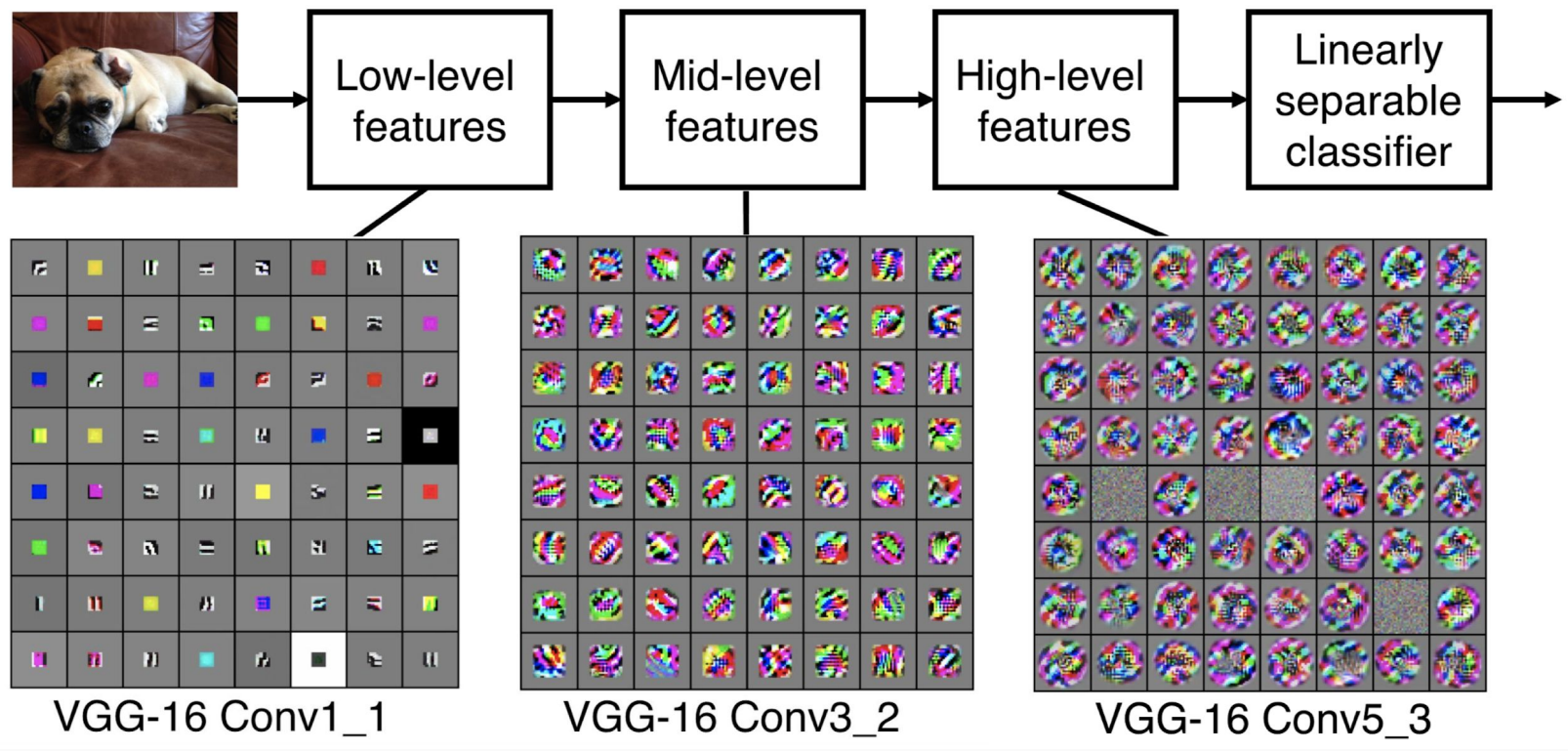
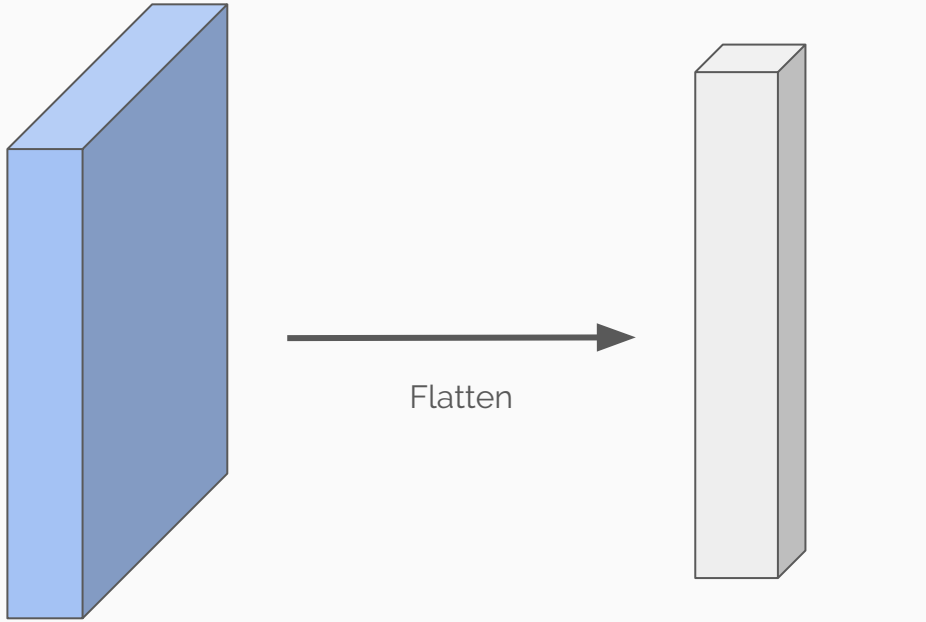


Image from CS231N Lecture by Andrej Karpathy and Visualization by Lane Mcintosh.

Convolutional Network (cont.)

Once you have enough high level features, you can flatten it to get a feature vector

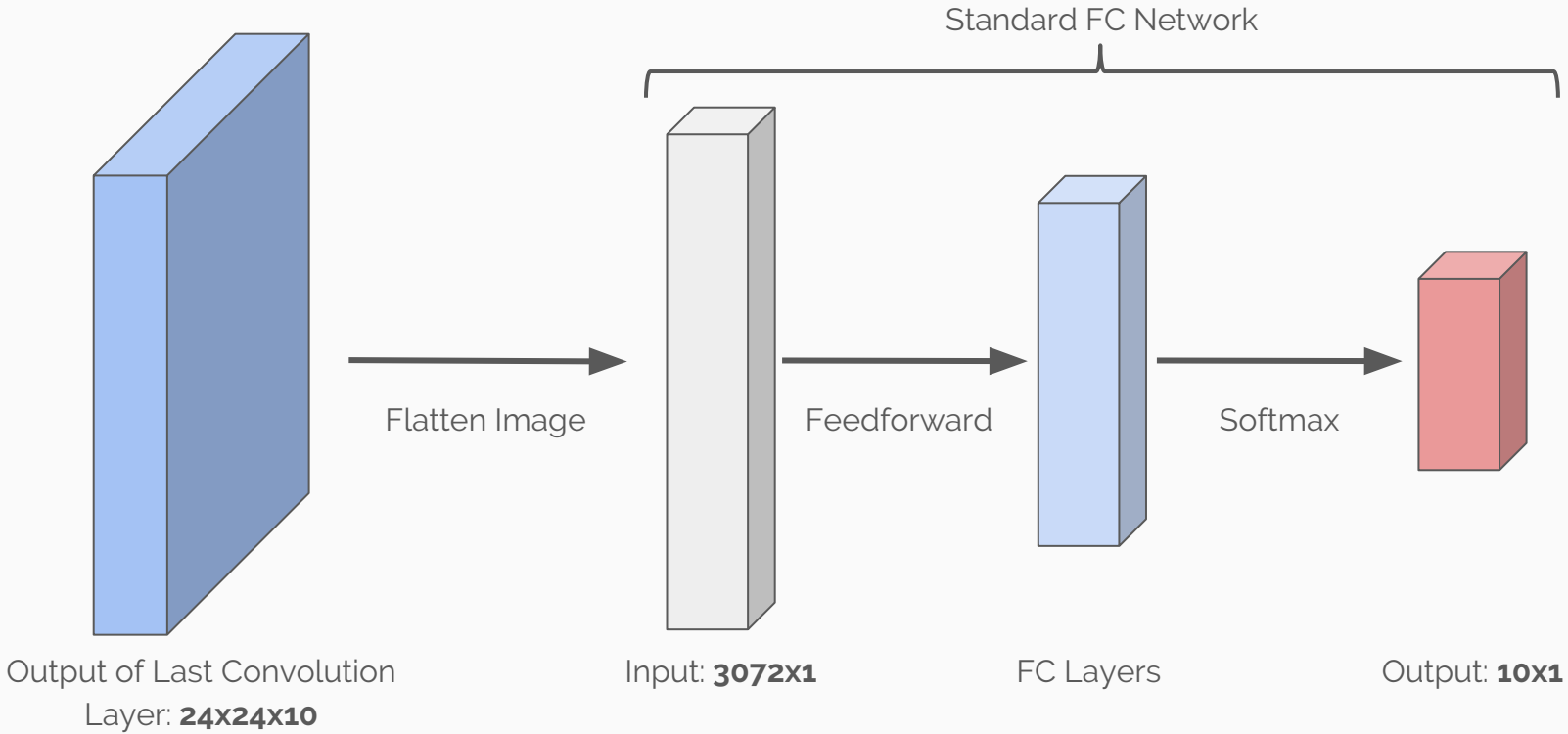


Output of Last Convolution Layer: **24x24x10**

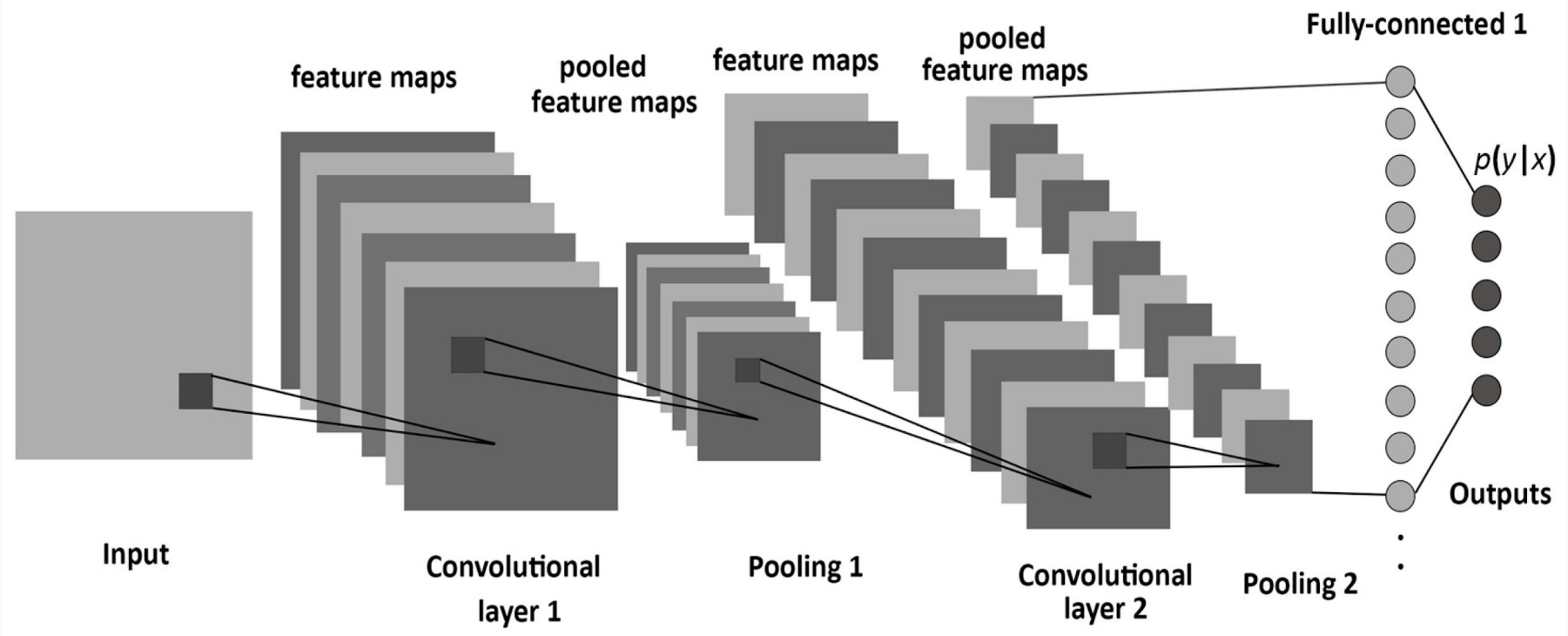
Feature Vector: **5760x1**

Convolutional Network (cont.)

We can then do the standard Fully Connected (FC) neural network prediction features



Convolutional Network (cont.)



Visualization of an example convolutional neural network

Standard Architectures

Trends towards smaller filters and deeper architectures

Trend towards removing Pooling and Fully-Connected layers

Architecture:

1. Input Image
2. Convolutional Layer 1 + Pooling + Activation (ReLU)
3. Convolutional Layer 2 + Pooling + Activation (ReLU)
4. …
5. Flatten Layer
6. Fully Connected Layers
7. Softmax Prediction

Announcements

Announcements

Practical 2 is due today by **11:59 p.m**.

No class next week for Spring Break. As an alternative, a short assignment is due next Friday.

Questions?