Computer Vision: Fall 2022 — Lecture 11

Dr. Karthik Mohan

Univ. of Washington, Seattle

November 4, 2022

Check-In

1. Identify your team mate through the spreadsheet

Check-In

1. Identify your team mate through the spreadsheet
2. First Check Point/Deadline for Mini-Project due November 6

Check-In

1. Identify your team mate through the spreadsheet
2. First Check Point/Deadline for Mini-Project due November 6
3. Fill out the mid-course survey! → General tab of Discord

References

1. Good Book for Machine Learning Concepts
2. Deep Learning Reference
3. Convolutional Neural Networks for Visual Recognition
4. Convolutional Neural Net Tutorial

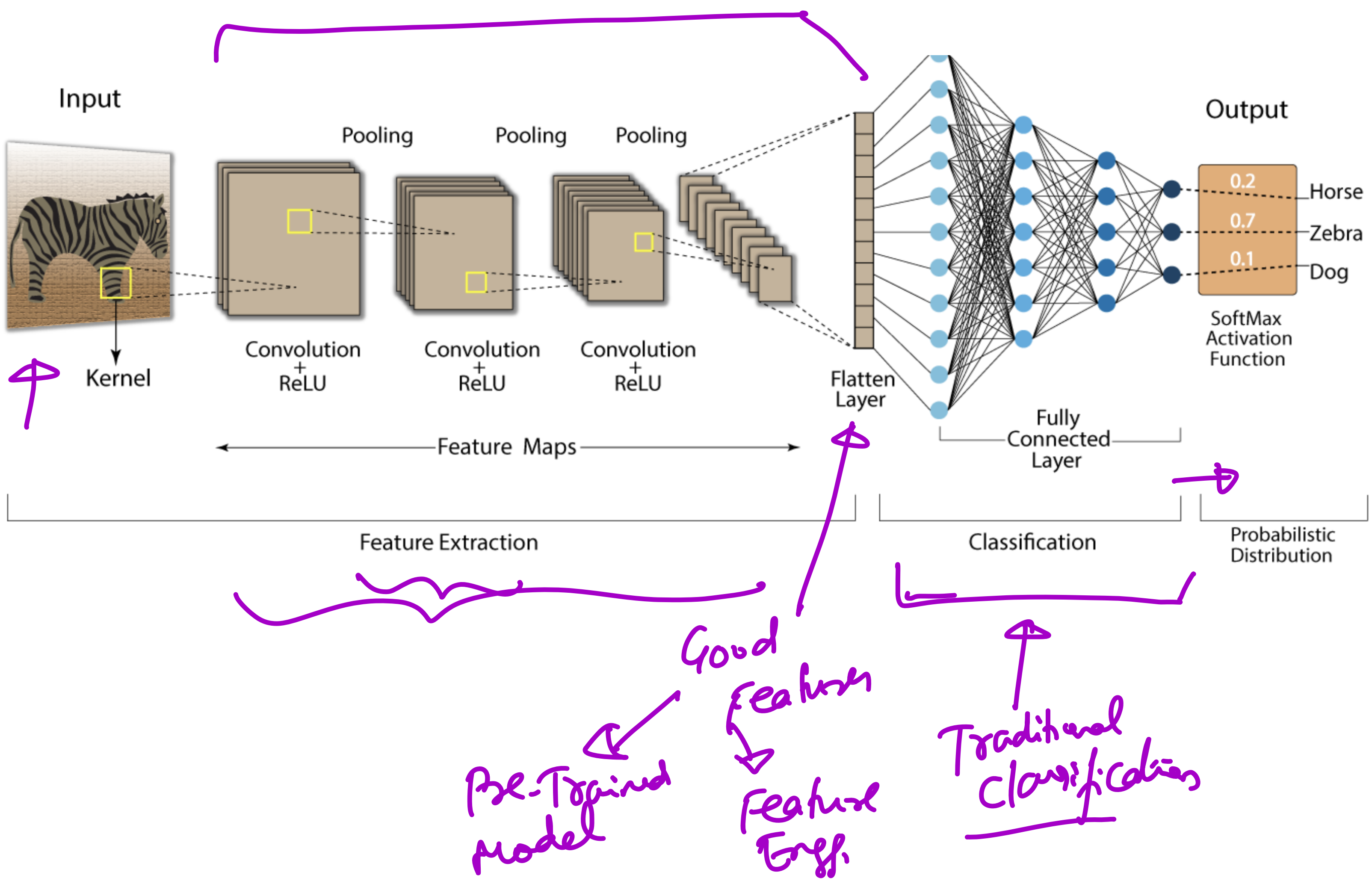
CNN Publication References

1. Convolutional Neural Networks: A comprehensive survey, 2019
2. A survey of Convolutional Neural Networks: Analysis, Applications, and Prospects, 2021
3. GoogLeNet
4. Top models on ImageNet

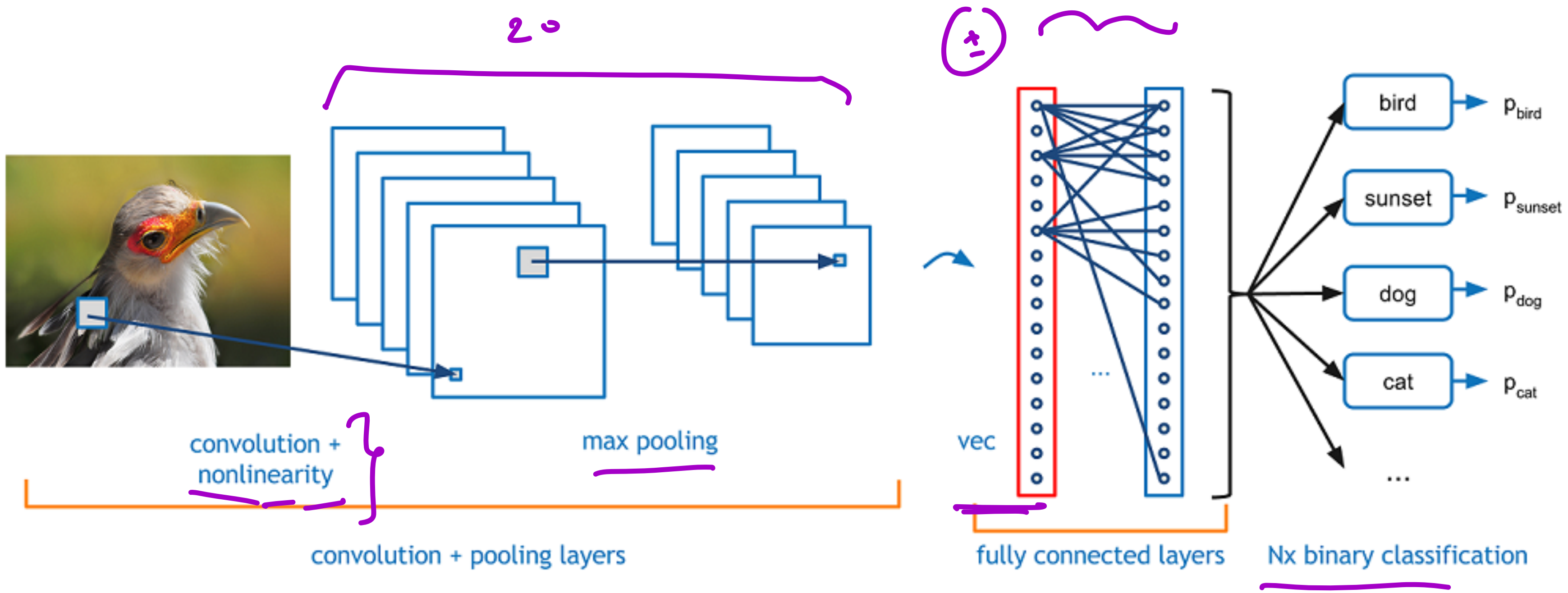
Today

1. Convolutional Neural Networks - Recap
2. CNN Architectures

Convolutional Neural Networks - Functionality Breakdown



Convolutional Neural Networks - Layers Breakdown



Types of Layers/Transforms in CNN

FC Layer

This is the same as in a feed-forward NN arch. Every neuron in the next layer is connected to every neuron in previous layer - Hence FC or *fully connected*

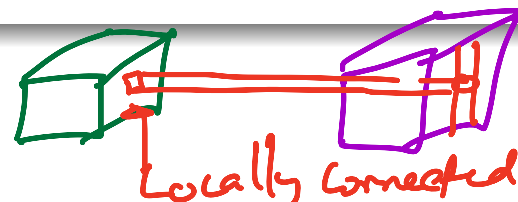
Types of Layers/Transforms in CNN

FC Layer

This is the same as in a feed-forward NN arch. Every neuron in the next layer is connected to every neuron in previous layer - Hence FC or *fully connected*

Conv Layer

(Locally connected Layer)



This is the most important and frequently used layer in a CNN arch - Here one or more Convolution Kernels (learned as parameters in training) are each convolved with the input to produce an output block with the same depth as the number of convolution kernels.

Types of Layers/Transforms in CNN

Pooling Layer (Locally connected)

Usually used to reduce the total number of parameters in the CNN network - Pooling can reduce the number of neurons from one layer to next with simple operations.

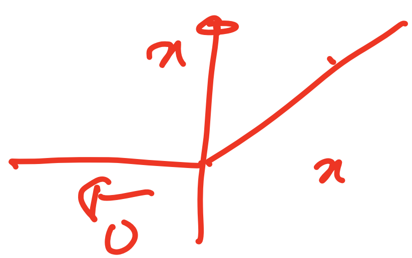
Types of Layers/Transforms in CNN

Pooling Layer

Usually used to reduce the total number of parameters in the CNN network - Pooling can reduce the number of neurons from one layer to next with simple operations.

RELU Activation

Just like in NN arch - RELU is used in CNN as well as a non-linear transformation of neurons.



Conv Layer

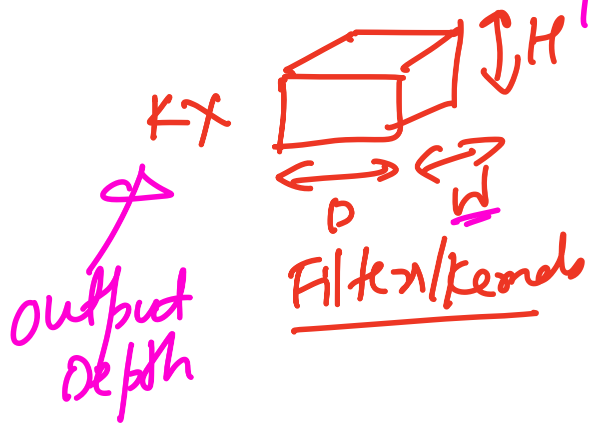
K kernels

Conv Layer Parameters

1. Convolution Kernel - has size $WxHxD$. Usually $3x3xD$ where D is the depth of the input. If the output block has a depth of K - This implies K such kernels are learned in that layer!

Parameters – $W \times H \times D$ } 1 kernel

total # parameters – $K \times (W \times H \times D)$ ← Input Depth



Conv Layer

Conv Layer Parameters

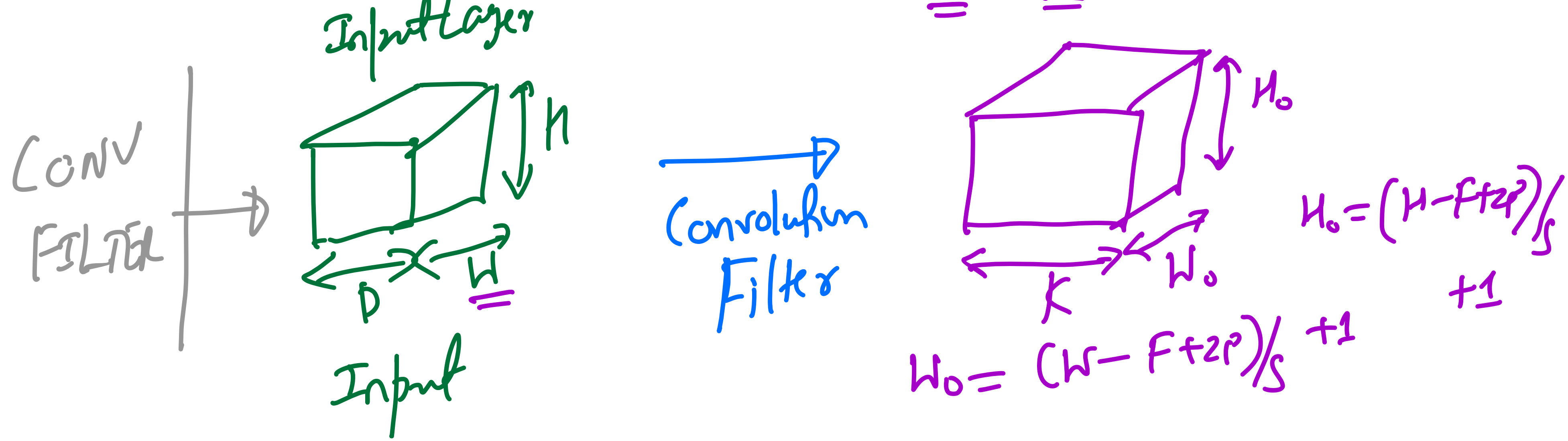
1. Convolution Kernel - has size $WxHxD$. Usually $3x3xD$ where D is the depth of the input. If the output block has a depth of K - This implies K such kernels are learned in that layer!

Conv Layer Hyper-Parameters

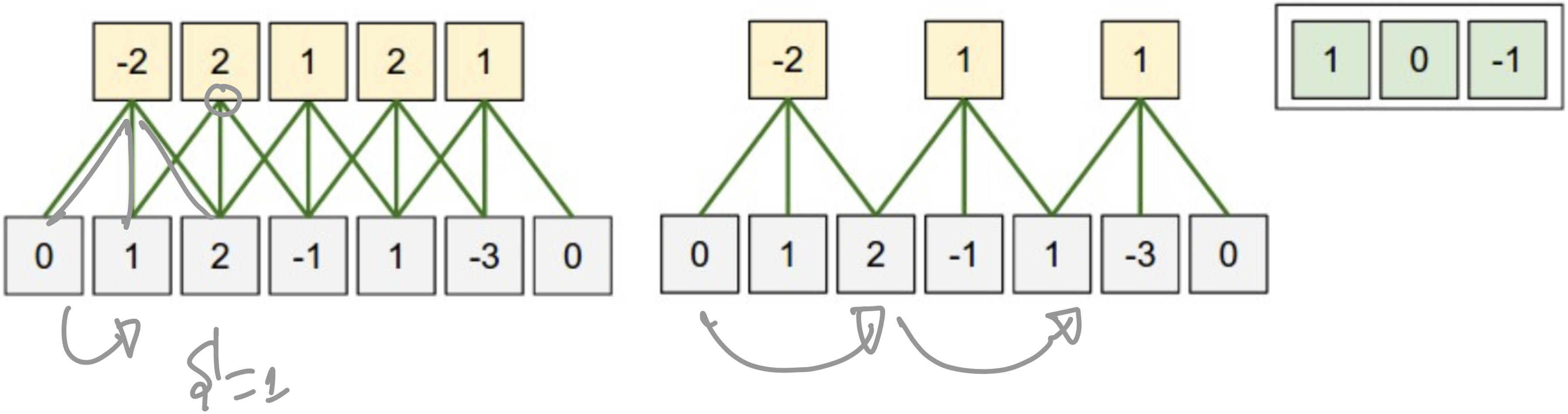
1. K or depth of the output block or the number of convolution kernels/filters
2. Stride Length, S: How much to shift the convolution kernel by when passing through the input
3. Zero-Padding, P: How much to pad the input before convolution (this impacts the output size!

Conv Layer

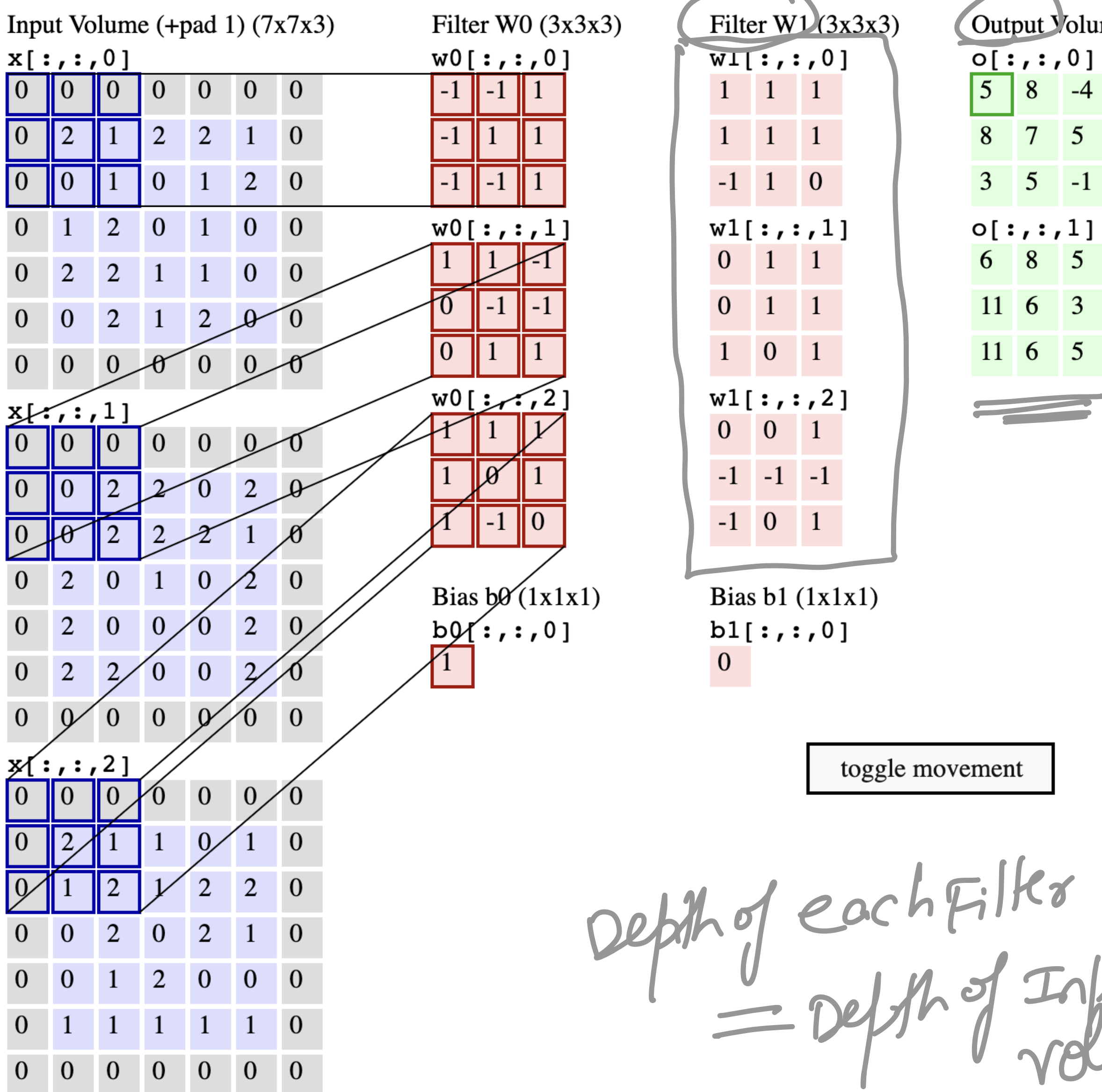
1. Let F be the receptive field size of the convolution Kernel
2. Let S be the stride length
3. Let P be the zero-padding
4. Width of the output block is now $(W - F + 2P)/S + 1$!



Convolution with Strides



Conv Layer Computations

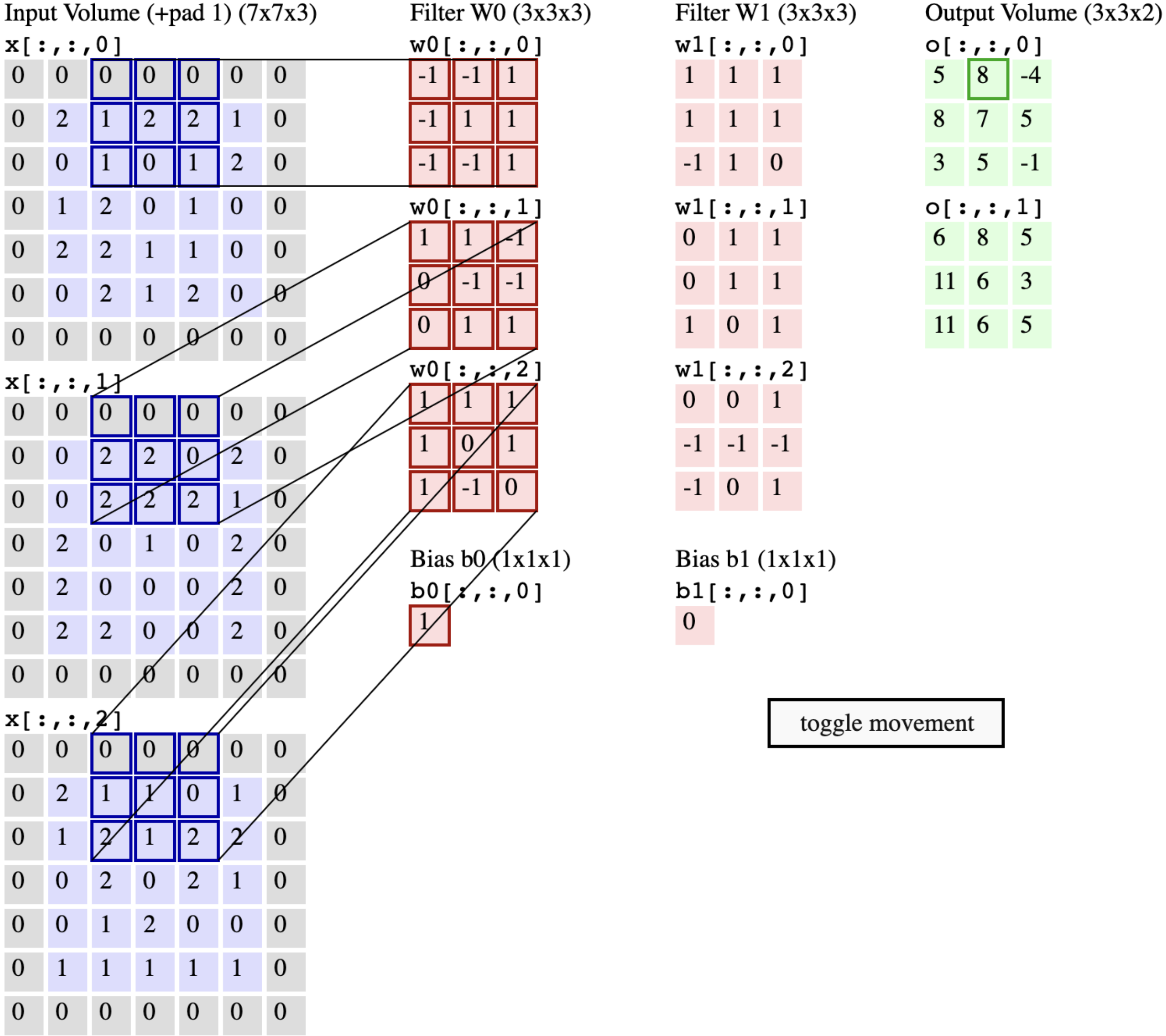


Depth = 1

Depth = 2

Depth of each Filter (3D conv/3D Filter)
= Depth of Input volume

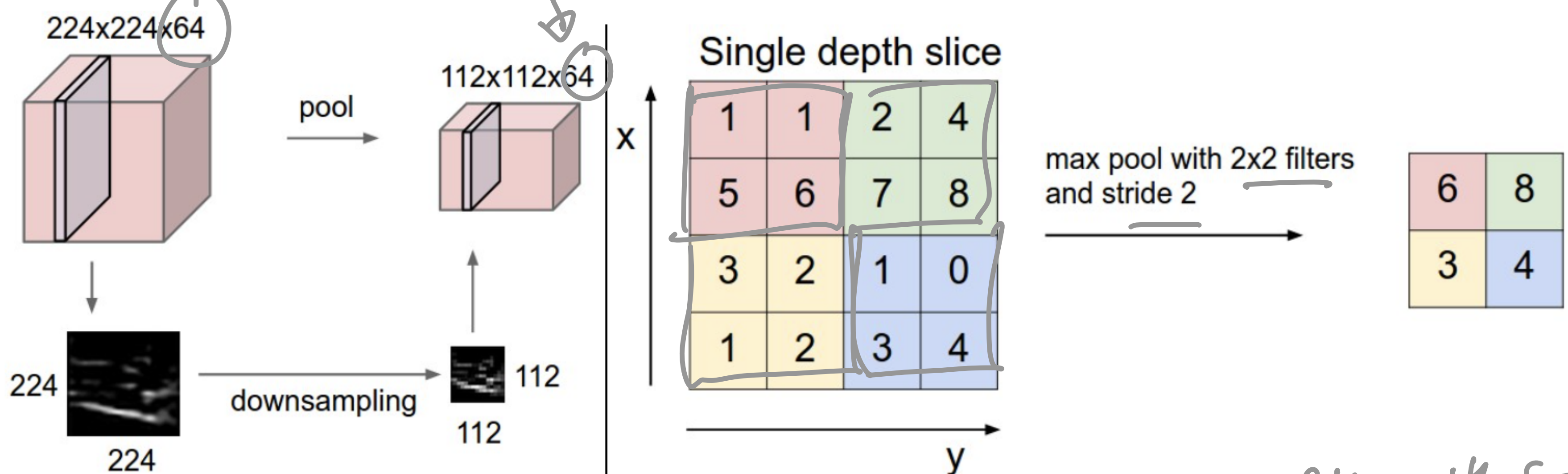
Conv Layer Computations



Conv Layer Computation Animation

Pooling Layer - Max Pooling Example

Depth is same



1. (Non-overlapping max-pool) → 2x2 filter with S=2

2. Max-pool with 2x2 filter & S=1

↳ overlapping maxpool

Pooling Layer

1. Reduces size of layers in CNN and hence reduces number of parameters

Pooling Layer

1. Reduces size of layers in CNN and hence reduces number of parameters
2. Usually $F = 2, S = 2$, i.e non-overlapping pooling with 2x2 size - Downsample each dimension by 2!

Pooling Layer

1. Reduces size of layers in CNN and hence reduces number of parameters
2. Usually $F = 2, S = 2$, i.e non-overlapping pooling with 2x2 size - Downsample each dimension by 2!
3. In pooling - Depth doesn't change from input to output layer. So pool across each depth slice. Contrast this with conv layer - where depth of output depends on the number of convolution kernels K, used!

Pooling Layer

1. Reduces size of layers in CNN and hence reduces number of parameters
2. Usually $F = 2, S = 2$, i.e non-overlapping pooling with 2x2 size - Downsample each dimension by 2!
3. In pooling - Depth doesn't change from input to output layer. So pool across each depth slice. Contrast this with conv layer - where depth of output depends on the number of convolution kernels K, used!
4. Pooling can be max or average - Max pooling works best!

Max Pooling

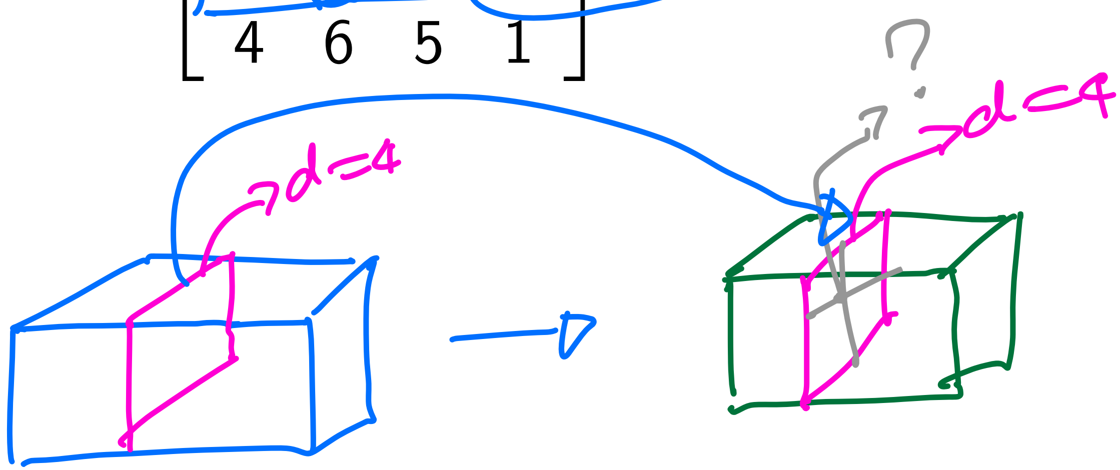
~~overlapping~~

Consider you are max pooling with $F = 2$ and stride length, $S = 1$ and a zreo padding, $P = 0$. Consider the input block, I at a *depth slice* of 4, i.e. an image matrix, I as below. What is the value at the second row, second column of the output block corresponding to this *depth slice* of 4?

$$I = \begin{bmatrix} 1 & 6 & 3 & 4 \\ 3 & 4 & 5 & 2 \\ 1 & 5 & 3 & 2 \\ 4 & 6 & 5 & 1 \end{bmatrix}$$

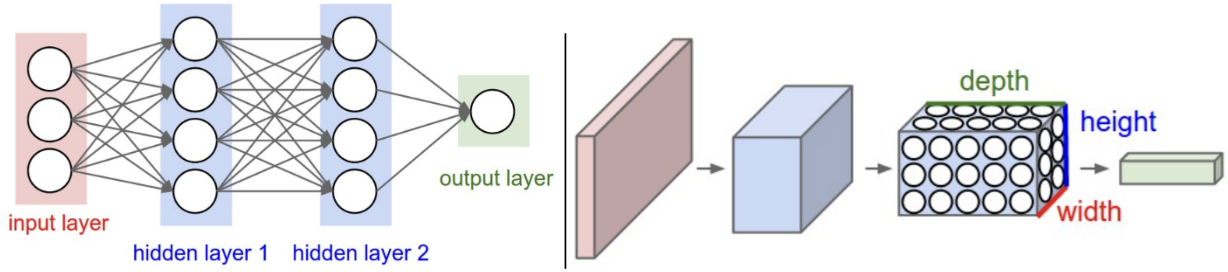
Row=1

2nd row, 2nd col



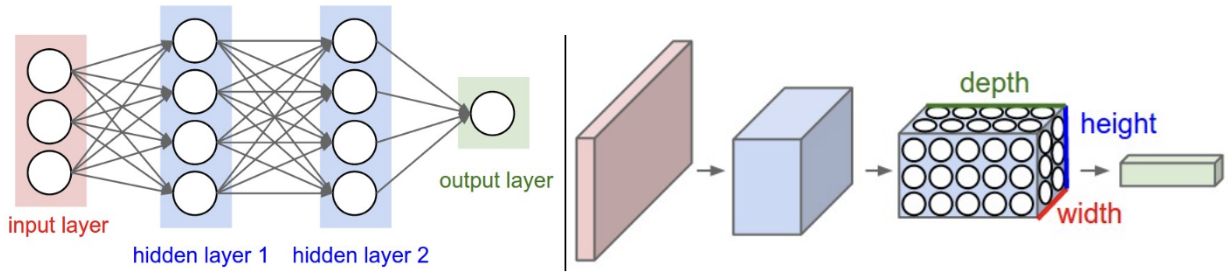
- (a) 3
- (b) 4
- (c) 5
- (d) 6

CNN vs NN



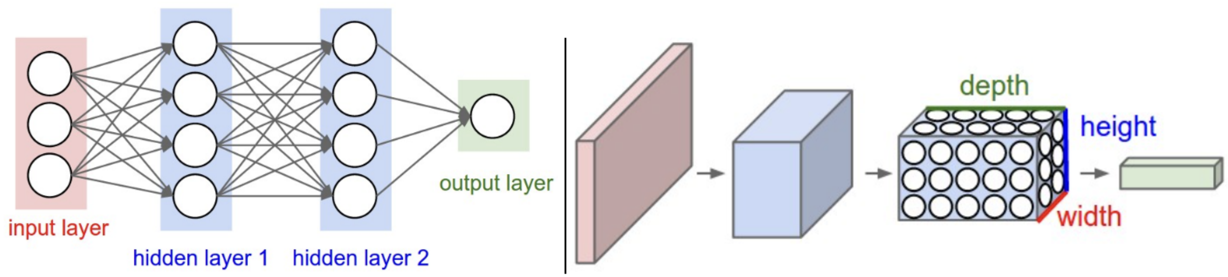
1. CNN is a special type of NN

CNN vs NN



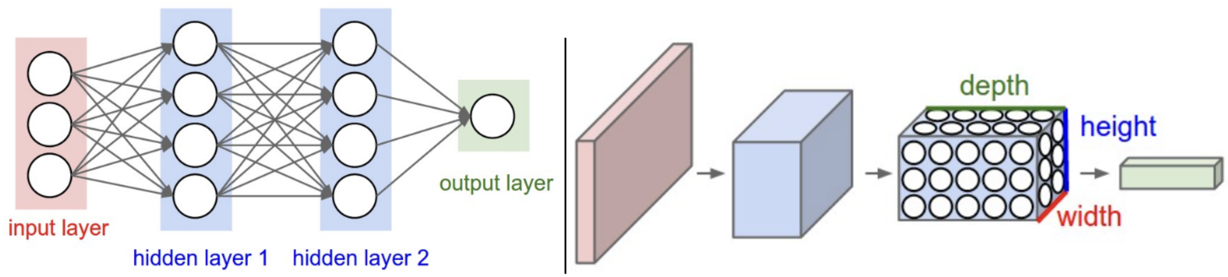
1. CNN is a special type of NN
2. Specialized to Images

CNN vs NN



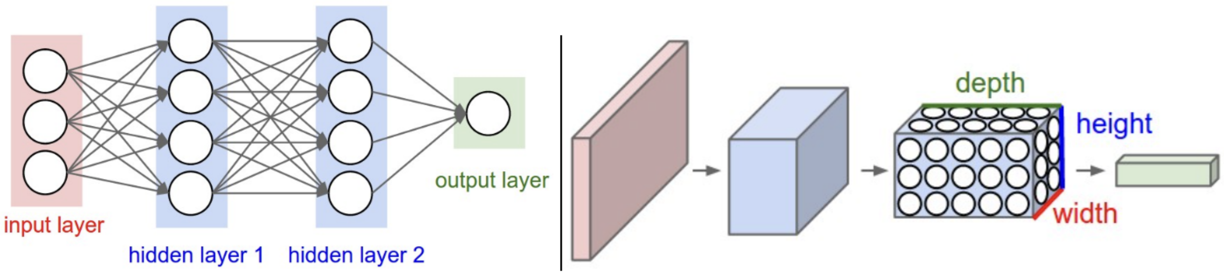
1. CNN is a special type of NN
2. Specialized to Images
3. More intuitive feature engineering (in terms of convolutions) done by CNN as compared to a regular NN

CNN vs NN



1. CNN is a special type of NN
2. Specialized to Images
3. More intuitive feature engineering (in terms of convolutions) done by CNN as compared to a regular NN
4. Works on a block with height, width and depth as compared to a NN, where the layers are encoded as vectors.

CNN vs NN



1. Fundamental unit in CNN is a block (with width, W, height H, and depth D). Fundamental unit in NN is a vector of neurons.
2. NN only has a feedforward connection (mostly) from one vector of neurons to another. CNN has 3 different types of connections - FC, Conv, and Pooling.
3. NN has full connectivity. CNN has local connectivity (e.g. conv Layer and Pooling)
4. Feedforward NN parameter space would be prohibitively large for Images. Conv Nets have shared parameter space and keep the parameter space manageable.

Next Topic: Popular CNN Architectures

Popular CNN Architectures

Arch	Year	Mention	Speciality
LeNet	1998	Yann LeCun et al	
AlexNet	2012	*Runner-up	Deeper, Bigger 8 % delta
ZFNet	2013	*Winner	Improvement on AlexNet
GoogLeNet	2014	*Winner	Inception Module 60 MM $\rightarrow$ 4 MM params
VGGNet	2014	*Runner-up	Deep network (16 layers) with 140 MM params
ResNet	2015	*Winner	Skip-connections and Batch-normalization

Table: Why competitions matter? *ILSVRC challenge (Evolution of CNN archs over the years)

Popular CNN Architectures

Year	CNN	Developed By	Error Rates	No. of Parameters	Dataset
1998	LeNet	Yann LeCun		60 Thousand	ImageNet
2012	AlexNet	Alex Krizhevsky, Geoffrey Hinton and Ilya Sutskever	15.3 %	60 Million	
2013	ZFNet	Matthew Zeiler, Rob Fergus	14.8 %		
2014	GoogleNet	Google	6.67 %	4 Million	
2014	VGGNet	Simonyan, Zisserman	7.3 %	138 Million	
2015	ResNet	Kaiming He	3.6 %		

ImageNet Data Set and ILSVRC

1. **ImageNet:** Launched in 2009 by Fei-Fei Li to have a large scale and clean image data set for benchmarking

ILSVRC Benchmark

ImageNet Data Set and ILSVRC

1. **ImageNet:** Launched in 2009 by Fei-Fei Li to have a large scale and clean image data set for benchmarking
2. **Classes:** 1000 classes

ILSVRC Benchmark

ImageNet Data Set and ILSVRC

1. **ImageNet:** Launched in 2009 by Fei-Fei Li to have a large scale and clean image data set for benchmarking
2. **Classes:** 1000 classes
3. **DataSet:** 1.3 MM training images, 50k validation and 1 MM test images

ILSVRC Benchmark

ImageNet Data Set and ILSVRC

1. **ImageNet:** Launched in 2009 by Fei-Fei Li to have a large scale and clean image data set for benchmarking
2. **Classes:** 1000 classes
3. **DataSet:** 1.3 MM training images, 50k validation and 1 MM test images
4. **ILSVRC** - Challenge on Image-Net to improve classification accuracy (ImageNet Large Scale Visual Recognition Challenge)

ILSVRC Benchmark

ImageNet Data Set and ILSVRC

1. **ImageNet:** Launched in 2009 by Fei-Fei Li to have a large scale and clean image data set for benchmarking
2. **Classes:** 1000 classes
3. **DataSet:** 1.3 MM training images, 50k validation and 1 MM test images
4. **ILSVRC** - Challenge on Image-Net to improve classification accuracy (ImageNet Large Scale Visual Recognition Challenge)
5. **Metric:** Top-k error rate. Is any of the models top k results the correct label?

Top-1
Top-5

True
Cat

Model
Dog, Bear, Cat, Cow, Tiger → 1

ILSVRC Benchmark

ImageNet Data Set and ILSVRC

1. **ImageNet:** Launched in 2009 by Fei-Fei Li to have a large scale and clean image data set for benchmarking
2. **Classes:** 1000 classes
3. **DataSet:** 1.3 MM training images, 50k validation and 1 MM test images
4. **ILSVRC** - Challenge on Image-Net to improve classification accuracy (ImageNet Large Scale Visual Recognition Challenge)
5. **Metric:** Top-k error rate. Is any of the models top k results the correct label?
6. **Current best** Top-1 accuracy at 90 % - COCA model.

2022

ILSVRC Benchmark

ImageNet Data Set and ILSVRC

1. **ImageNet:** Launched in 2009 by Fei-Fei Li to have a large scale and clean image data set for benchmarking
2. **Classes:** 1000 classes
3. **DataSet:** 1.3 MM training images, 50k validation and 1 MM test images
4. **ILSVRC** - Challenge on Image-Net to improve classification accuracy (ImageNet Large Scale Visual Recognition Challenge)
5. **Metric:** Top-k error rate. Is any of the models top *k* results the correct label?
6. **Current best** Top-1 accuracy at 90 % - COCA model.
7. **Current best** Top-5 accuracy at 99 % - Florence-CoSwim-H model

ILSVRC Benchmark

ImageNet Data Set and ILSVRC

1. **ImageNet:** Launched in 2009 by Fei-Fei Li to have a large scale and clean image data set for benchmarking
2. **Classes:** 1000 classes
3. **DataSet:** 1.3 MM training images, 50k validation and 1 MM test images
4. **ILSVRC** - Challenge on Image-Net to improve classification accuracy (ImageNet Large Scale Visual Recognition Challenge)
5. **Metric:** Top-k error rate. Is any of the models top *k* results the correct label?
6. **Current best** Top-1 accuracy at 90 % - COCA model.
7. **Current best** Top-5 accuracy at 99 % - Florence-CoSwim-H model
8. **Meta Pseudo-Labels** performs well on both Top-1 and Top-5 accuracy

2019

ILSVRC Benchmark

ICE #2

Top k accuracy metric

Suppose you trained your favorite CNN model based on one of these archs (say VGGnet). Your model predicts the top 5 results for each of the following examples as follows:

True Label	Top 5 Predictions
Cat	{Cat, Dog, Mouse, Rabbit, Tiger }
Dog	{Cat, Mouse, Rabbit, Dog, Tiger }
Rabbit	{ Rabbit, Dog, Mouse, Tiger, Cat }
Bear	{ Dog, Cat, Rabbit, Tiger, Mouse }
Tiger	{ Cat, Dog, Tiger, Rabbit, Bear }

Table: 5 Test Examples

ICE #2

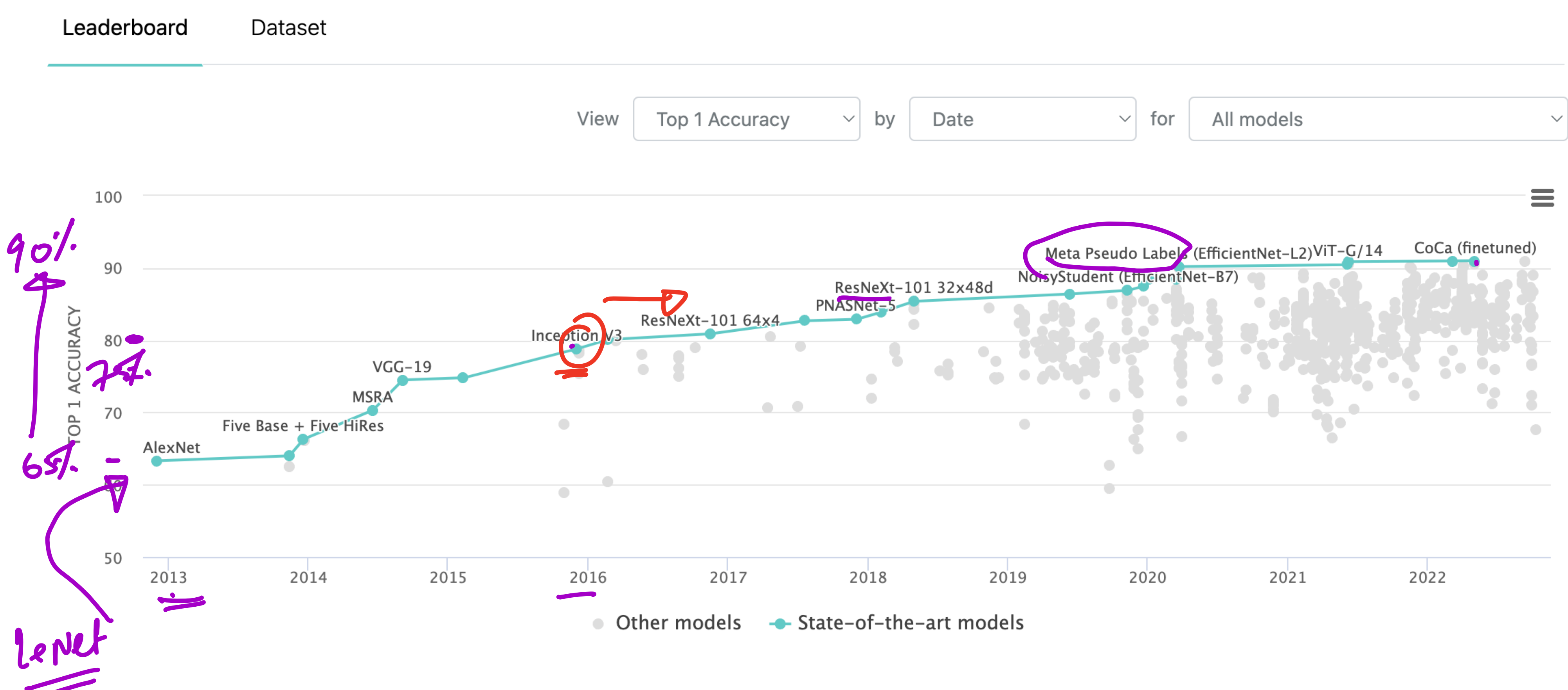
True Label	Top 5 Predictions
Cat	{Cat, Dog, Mouse, Rabbit, Tiger }
Dog	{Cat, Mouse, Rabbit, Dog, Tiger }
Rabbit	{ Rabbit, Dog, Mouse, Tiger, Cat }
Bear	{ Dog, Cat, Rabbit, Tiger, Mouse }
Tiger	{ Cat, Dog, Tiger, Rabbit, Bear }

Table: 5 Test Examples

What's the Top-1 and Top-5 accuracy scores averaged over these 5 examples?

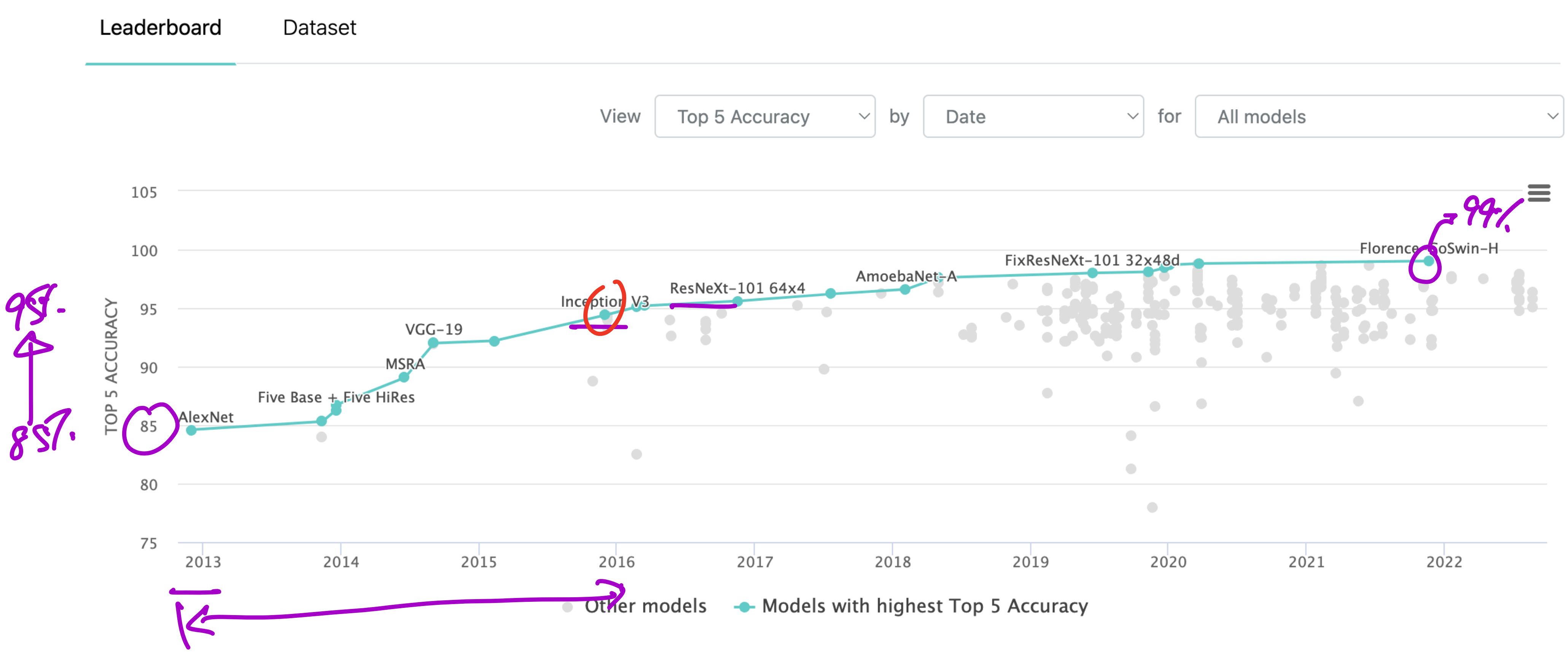
a. 40 % and 60 %
b. 60 % and 60 %
c. 40 % and 75 %
d. 40 % and 80 %

Top-1 Accuracy Evolution



Top models on ImageNet

Top-5 Accuracy Evolution



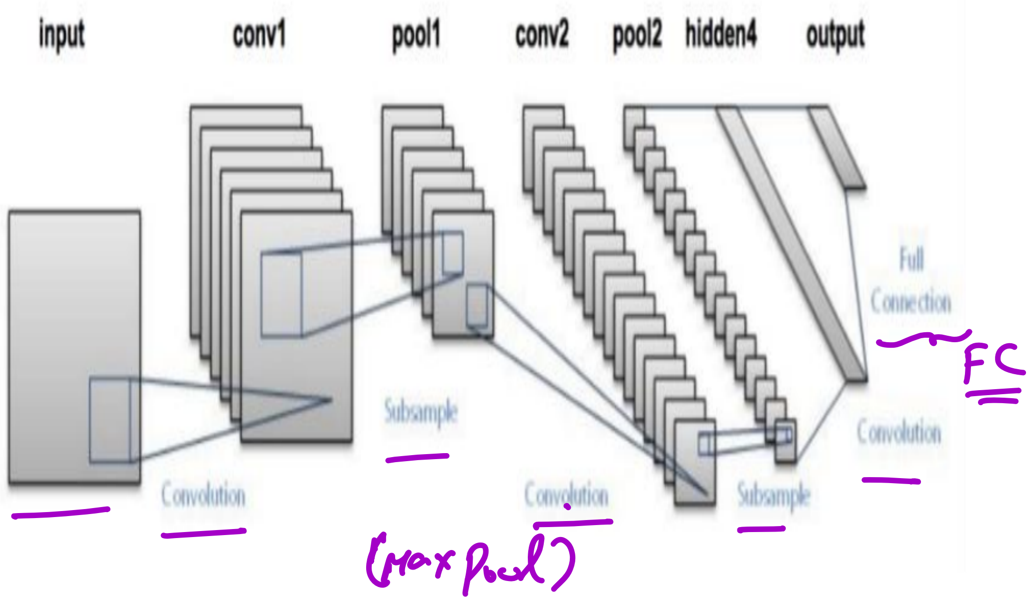
Top models on ImageNet

Popular CNN Architectures

Year	CNN	Developed By	Error Rates	No. of Parameters	Dataset
1998	LeNet	Yann LeCun		60 Thousand	ImageNet
2012	AlexNet	Alex Krizhevsky, Geoffrey Hinton and Ilya Sutskever	15.3 %	60 Million	
2013	ZFNet	Matthew Zeiler, Rob Fergus	14.8 %		
2014	GoogleNet	Google	6.67 %	4 Million	
2014	VGGNet	Simonyan, Zisserman	7.3 %	138 Million	
2015	ResNet	Kaiming He	3.6 %		

LeNet

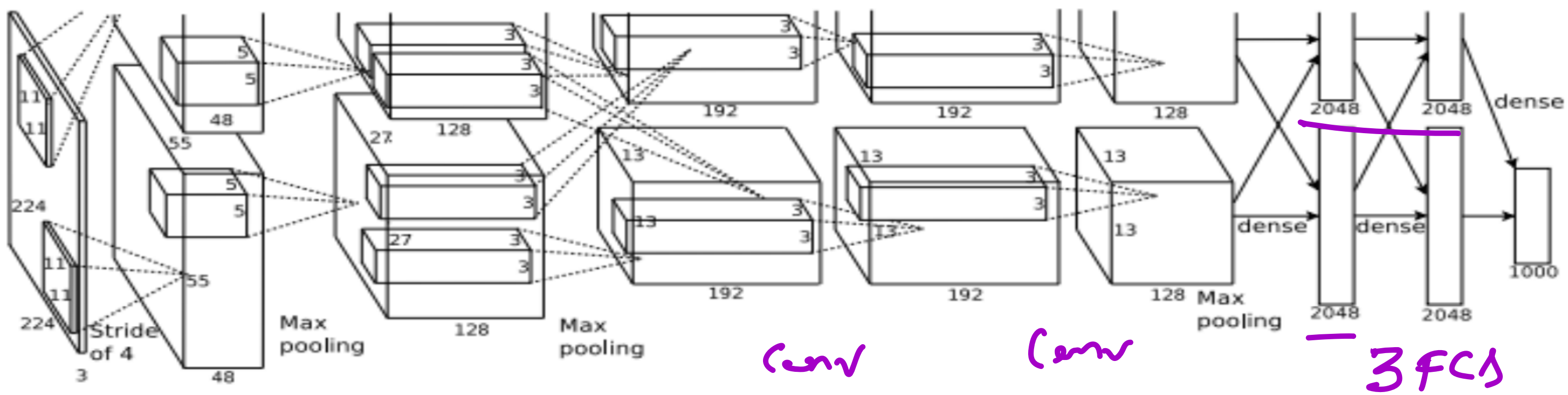
Yann Le-Cun
1999



LeNet → 3 Conv, 1 FC

AlexNet

~~2013~~ ILSVRC Winner
2012

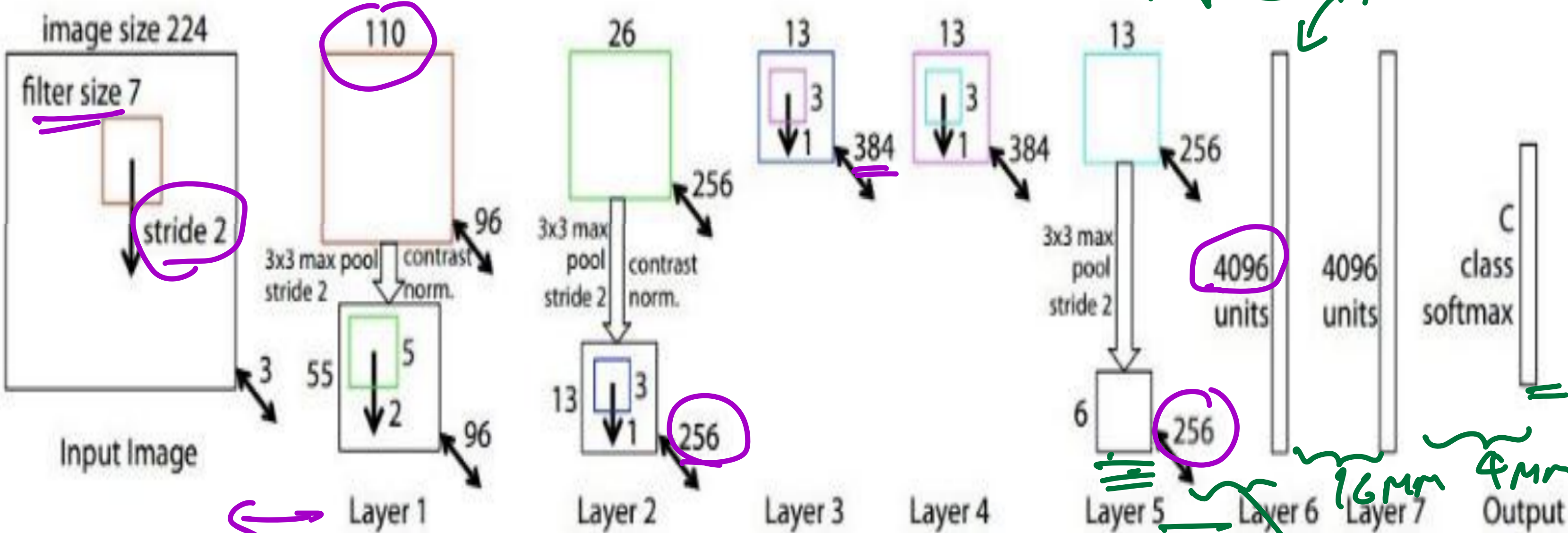


Conv Conv Conv 3 FCs

1. Incorporates RELU
2. Deeper layers than LeNet
3. Developed to measure lateral distance between vehicles

ZFNet

2013 winner



$(7\times7\times3)\times110 \approx O(10^3)$

$(3\times3\times256)\times384 = O(10^5)$ < 1MM

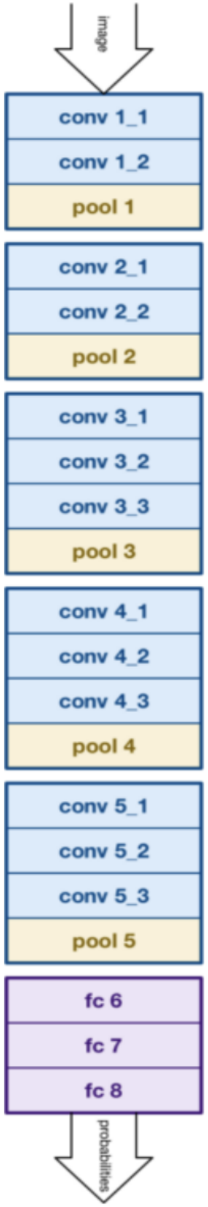
$(256\times6\times6) \times 4096 = O(28MM)$
depth

parameters
1MM +
30MM +
16MM +
4MM = 51MM

1. Hyper-parameter Tweaking in AlexNet
2. Small changes in structure
3. Number of params same as AlexNet: 60MM!
4. Top 5 Accuracy at 85.3% up from 84.6% of AlexNet

VGGNet

Runner-up for 2014 ILSVRC



Expansion on AlexNet/ZFNet

Stack conv Layers before maxpool

VGGNet

1. Top 5 Accuracy at 92% of VGGNet, up from 85.3% of ZFNet!
2. Runner up in the 2014 competition
3. Number of params: 138*MM*, up from 60*MM* of ZFNet!
4. Quite popular for image embeddings and representations
5. Prone to over-fit - Obviously!
6. *Applications:* Finger-print biometric authentication, crack detection, object tracking.

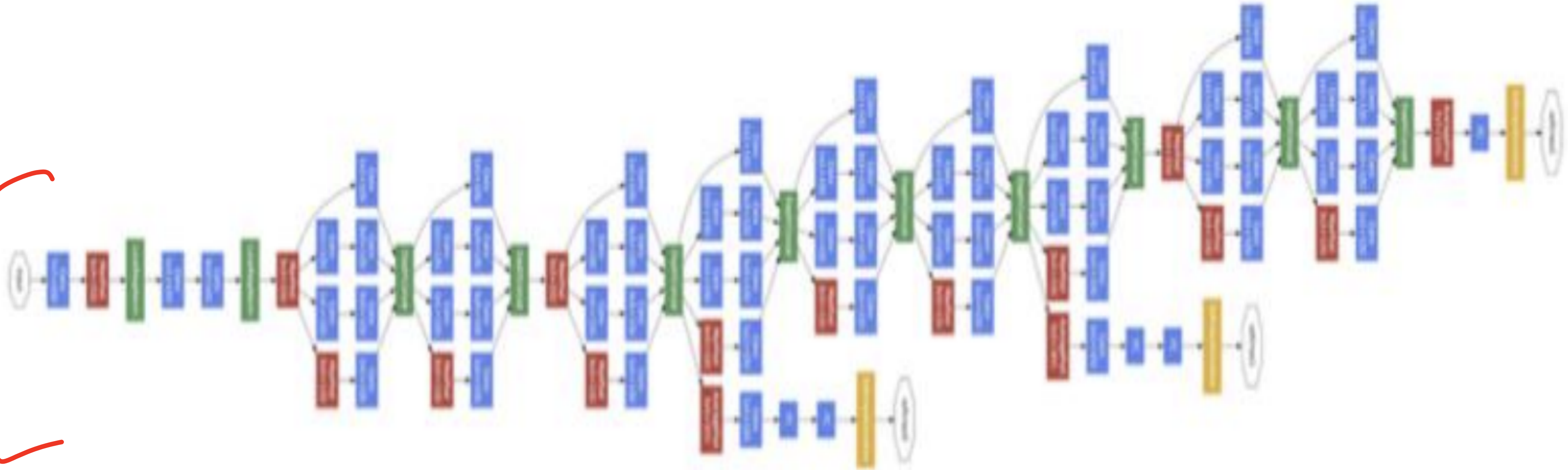
Inception/GoogleNet Motivation

Winner 2014 ILSVRC

(a) Siberian husky

(b) Eskimo dog

Inception/GoogLeNet



#Layers = 22

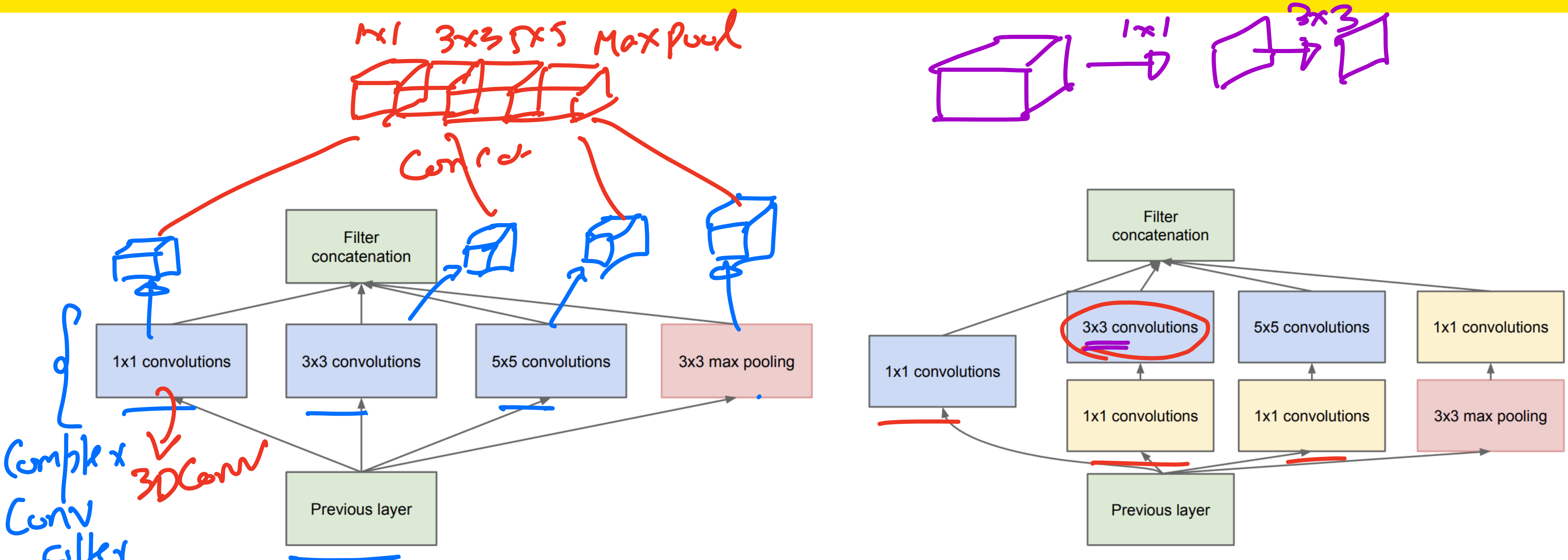
Inception

Inception/GoogLeNet

85.3% of ZFnet

1. Top 5 Accuracy at 94.4% up from ~~92%~~ of VGGNet
2. Introduced an Inception Module
3. Has many more layers than AlexNet or ZFNet!
4. 22 layers deep!
5. Number of params: $4MM$, down from $60MM$ of ZFNet!

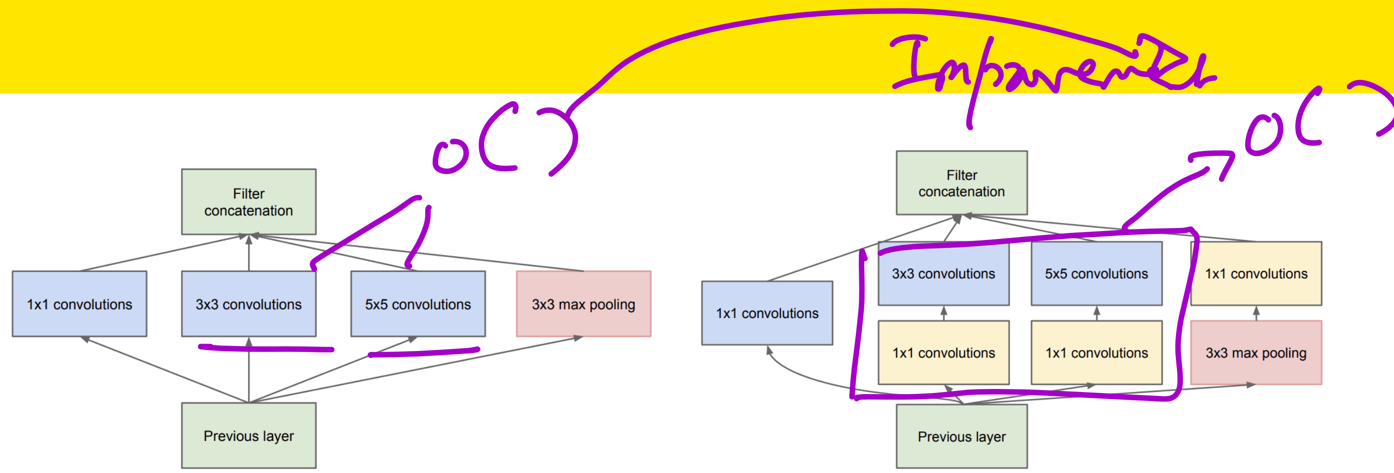
Inception Module



(a) Inception module, naïve version (b) Inception module with dimension reductions

1. Concatenates the depth from each of the convolutions
2. Allows for looking at the input at different scales (1x1, 3x3, 5x5, etc)
3. Let's the model use information from all scales

ICE #3



(a) Inception module, naïve version (b) Inception module with dimension reductions

$\frac{9+25}{2} = 17$

Inception!

$(w \times h) \times ((3 \times 3 \times d) + (5 \times 5 \times d))$ $(w \times h) \times (1 \times 1 \times d) \times 2 + 9 \times w \times h \times 1 + 25 \times w \times h \times 1$

Consider the inception Module used by GoogleNet as one of its layers. Between the reduced dimension version on the right and the one on the left - By what constant factor is the computational complexity reduced for the combination of the 3x3 and 5x5 conv layers inside the module?

1. 25
2. 35
3. 45
4. 55

Inception/GoogleNet Breakdown

Inception → 1x1 3x3 5x5 maxpool

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	$\boxed{128}$	128	$\boxed{192}$	32	$\boxed{96}$	$\boxed{64}$	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

4M

Inception Visual walkthrough

Next Lecture

1. ResNet
2. ResNeXt