



# CS489/698: Intro to ML

## Lecture 13: CNNs



# Outline

- Review
- Pooling
- Architectures



# Convolutional Layer

**Summary.** To summarize, the Conv Layer:

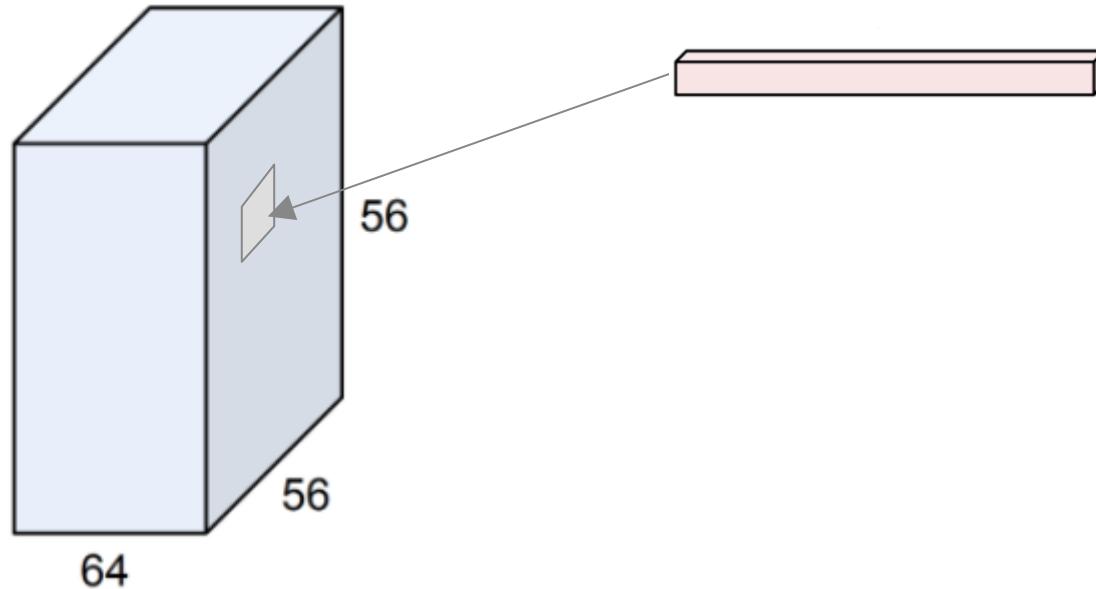
- Accepts a volume of size  $W_1 \times H_1 \times D_1$
- Requires four hyperparameters:
  - Number of filters  $K$ ,
  - their spatial extent  $F$ ,
  - the stride  $S$ ,
  - the amount of zero padding  $P$ .
- Produces a volume of size  $W_2 \times H_2 \times D_2$  where:
  - $W_2 = (W_1 - F + 2P)/S + 1$
  - $H_2 = (H_1 - F + 2P)/S + 1$  (i.e. width and height are computed equally by symmetry)
  - $D_2 = K$
- With parameter sharing, it introduces  $F \cdot F \cdot D_1$  weights per filter, for a total of  $(F \cdot F \cdot D_1) \cdot K$  weights and  $K$  biases.
- In the output volume, the  $d$ -th depth slice (of size  $W_2 \times H_2$ ) is the result of performing a valid convolution of the  $d$ -th filter over the input volume with a stride of  $S$ , and then offset by  $d$ -th bias.

**focal**  
Systems

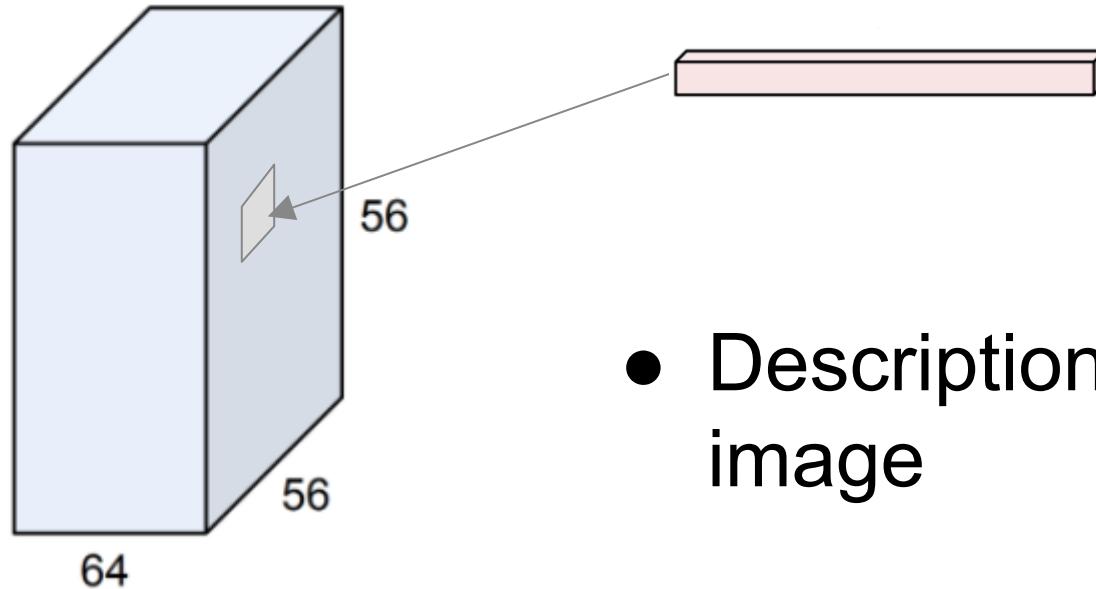


UNIVERSITY OF  
**WATERLOO**

# What is a conv feature?



# What is a conv feature?



- Description of region of image

# Motivation

- We care more about knowing that a feature is present than exactly where it is
- i.e. We don't need the exact pixels of Obama's eyes and nose to know it is obama's face



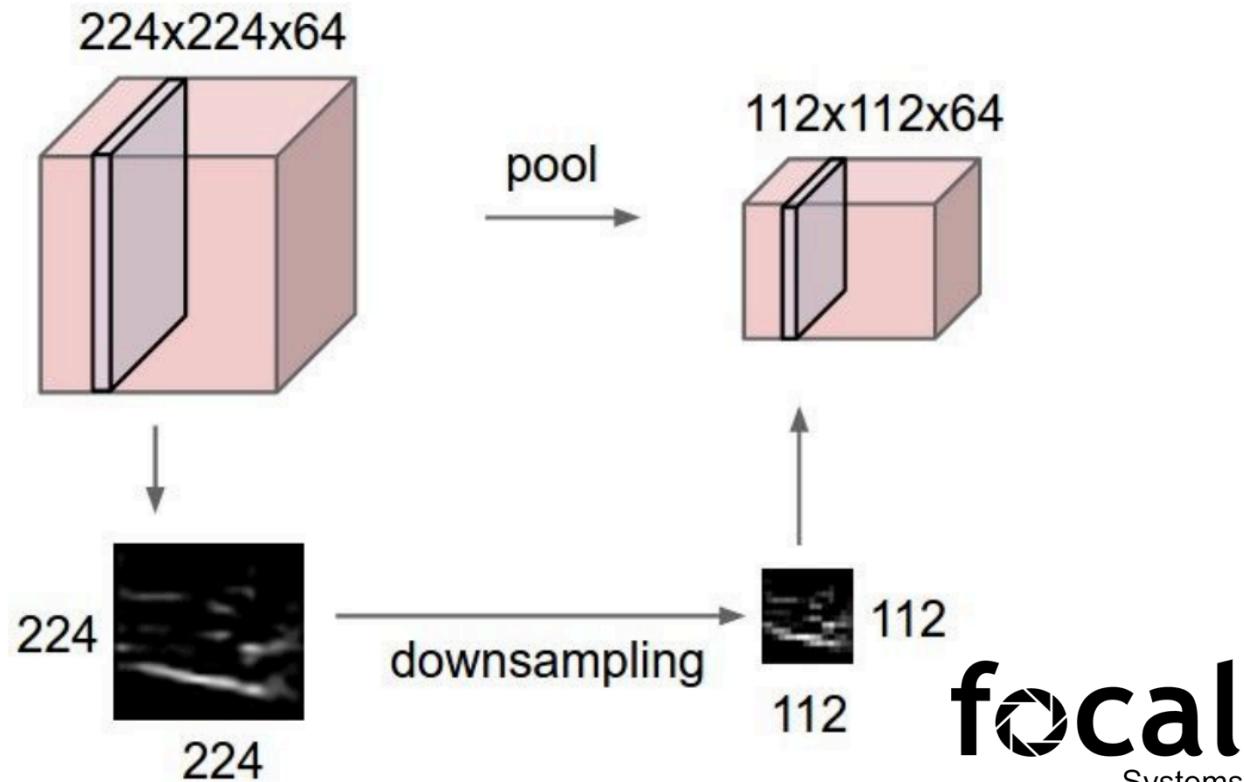
**focal**  
Systems



UNIVERSITY OF  
**WATERLOO**

# Solution: Pooling

- Replaces activations with summary statistic of nearby activations



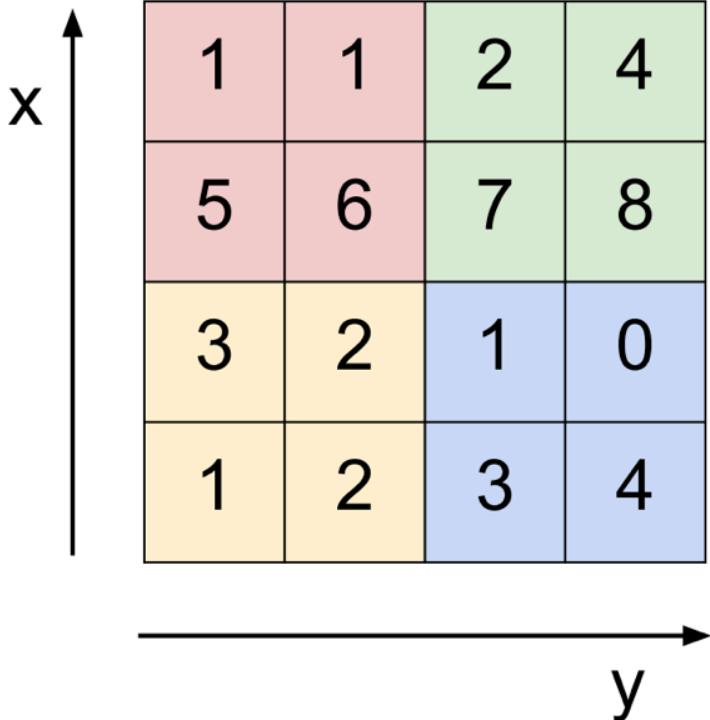
**focal**  
Systems



UNIVERSITY OF  
**WATERLOO**

# Example: Max Pooling

Single depth slice



max pool with 2x2 filters  
and stride 2



6	8
3	4

**focal**  
Systems



UNIVERSITY OF  
**WATERLOO**

# Benefits Of Pooling

- Some translation invariance
- No parameters
- Easy to backprop
- Less computations
- Increased receptive field

**focal**  
Systems



UNIVERSITY OF  
**WATERLOO**

# Where is pooling used?

- After series of convs
- Followed by double depth

ConvNet Configuration						
A	A-LRN	B	C	D	E	
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers	
input (224 × 224 RGB image)						
conv3-64	conv3-64 LRN	conv3-64 <b>conv3-64</b>	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool						
conv3-128	conv3-128	conv3-128 <b>conv3-128</b>	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool						
conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 <b>conv1-256</b>	conv3-256 conv3-256 <b>conv3-256</b>	conv3-256 conv3-256 <b>conv3-256</b>	conv3-256 conv3-256 conv3-256 <b>conv3-256</b>
maxpool						
conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool						
conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool						
FC-4096						
FC-4096						
FC-1000						
soft-max						

The 6 different architectures of VGG Net. Configuration D produced the best results

# Drawbacks of Pooling

- Loss of information

**focal**  
Systems



UNIVERSITY OF  
**WATERLOO**

# Pooling Layer Summary

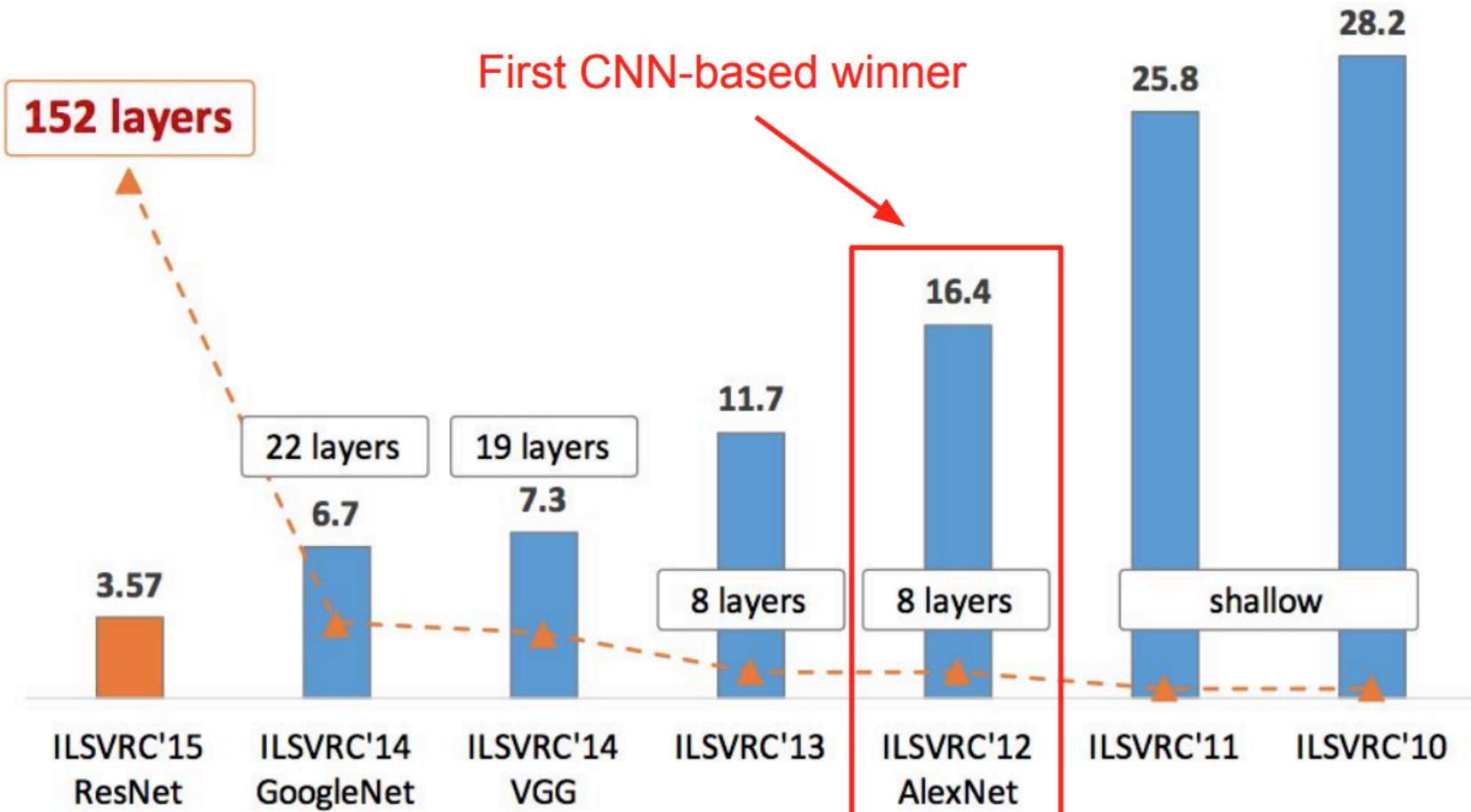
- Accepts a volume of size  $W_1 \times H_1 \times D_1$
- Requires three hyperparameters:
  - their spatial extent  $F$ ,
  - the stride  $S$ ,
- Produces a volume of size  $W_2 \times H_2 \times D_2$  where:
  - $W_2 = (W_1 - F)/S + 1$
  - $H_2 = (H_1 - F)/S + 1$
  - $D_2 = D_1$
- Introduces zero parameters since it computes a fixed function of the input
- Note that it is not common to use zero-padding for Pooling layers

**focal**  
Systems



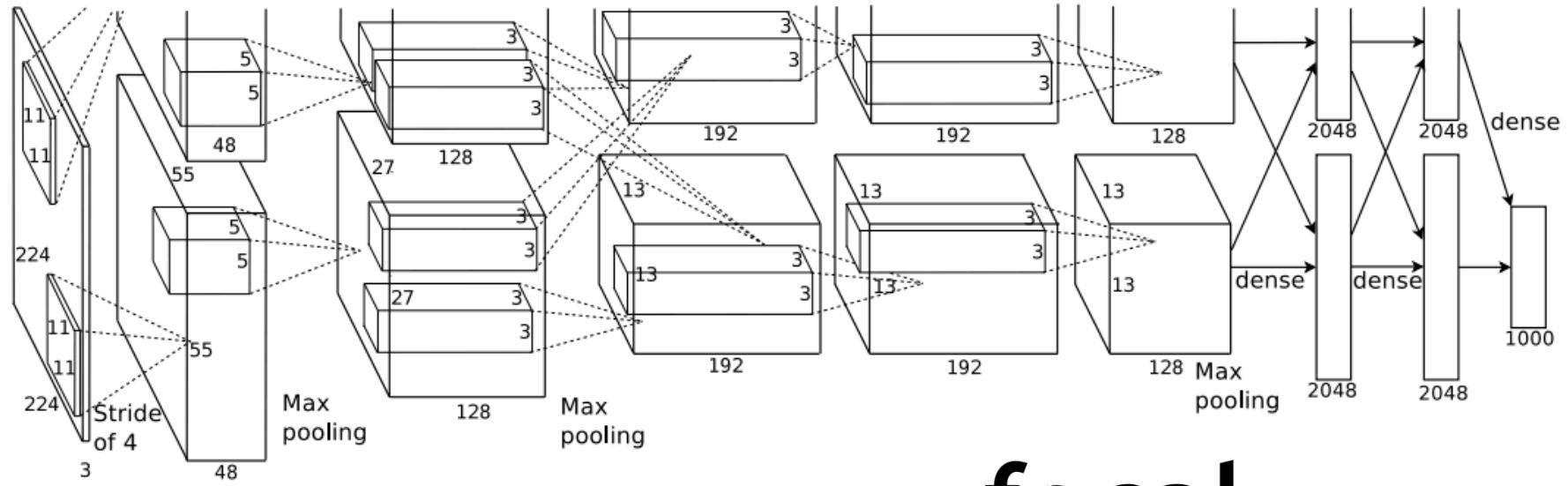
UNIVERSITY OF  
**WATERLOO**

# Architectures for ImageNet



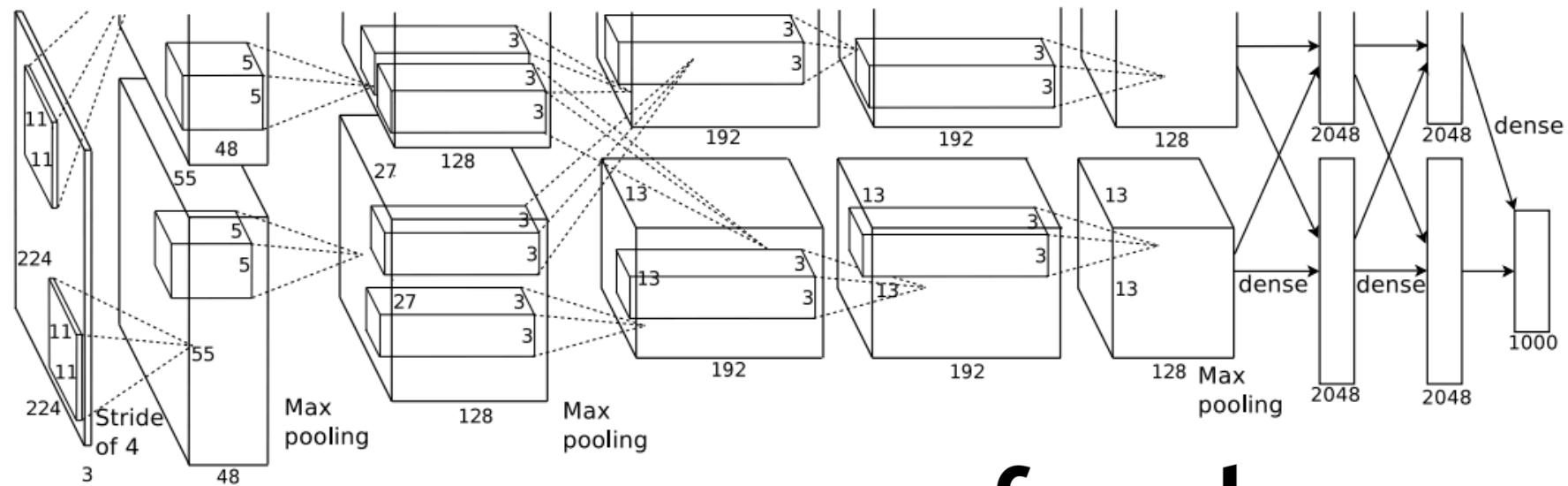
# AlexNet - 2012

- Input Image 227x227x3



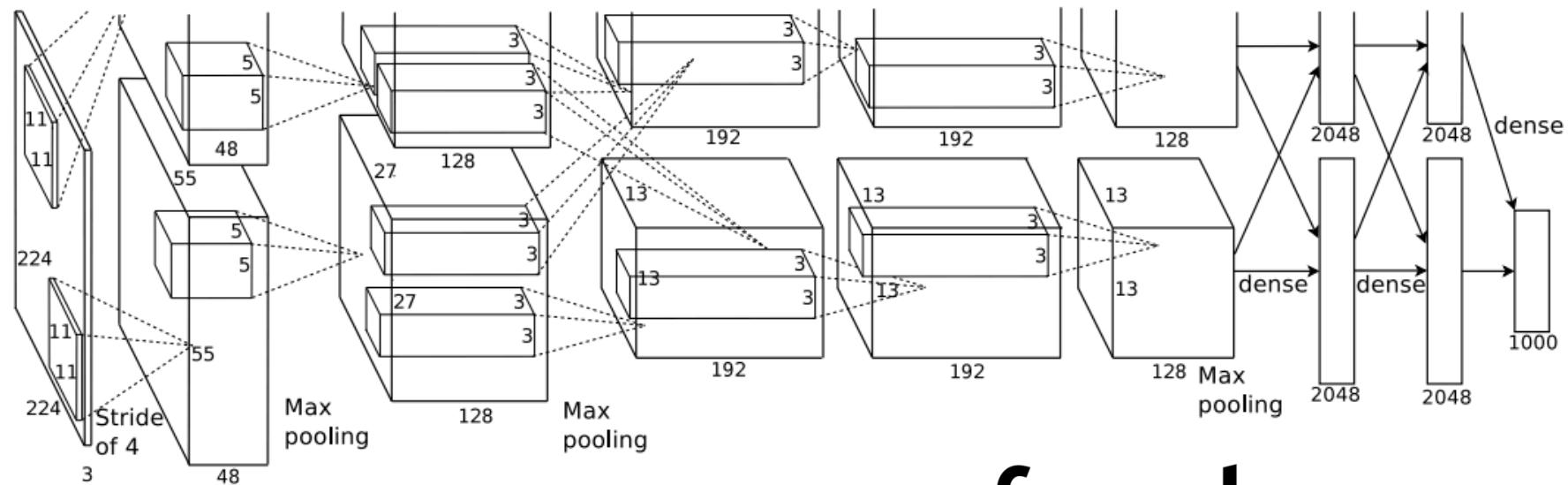
# AlexNet - 2012

- Input Image 227x227x3
- First Layer: 96 11x11 Conv filters stride 4 pad 0
- What is output size?



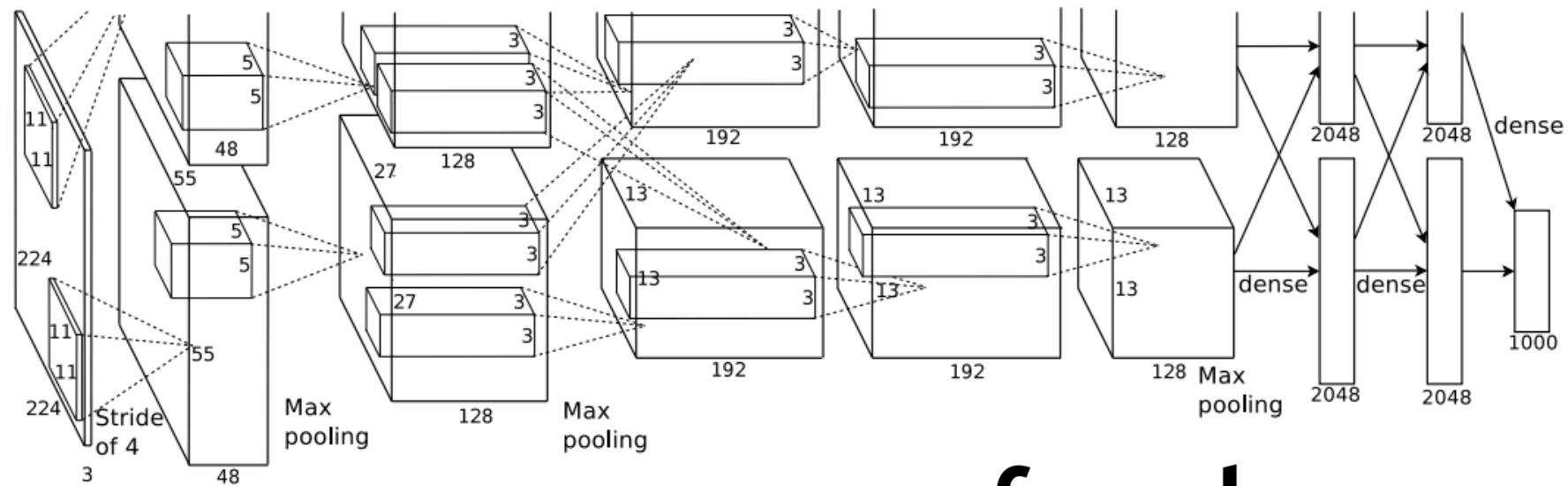
# AlexNet - 2012

- Input Image 227x227x3
- First Layer: 96 11x11 Conv filters stride 4 pad 0
- What is output size? 55x55x96



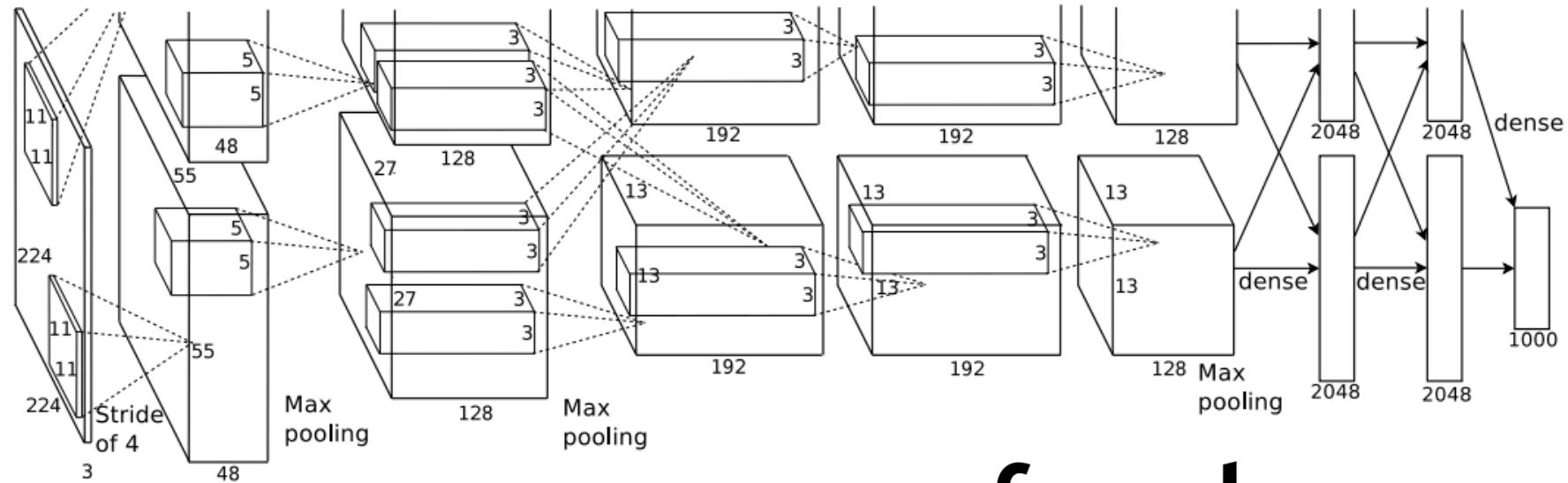
# AlexNet - 2012

- Input Image 227x227x3
- First Layer: 96 11x11 Conv filters stride 4 pad 0
- How Many Parameters?



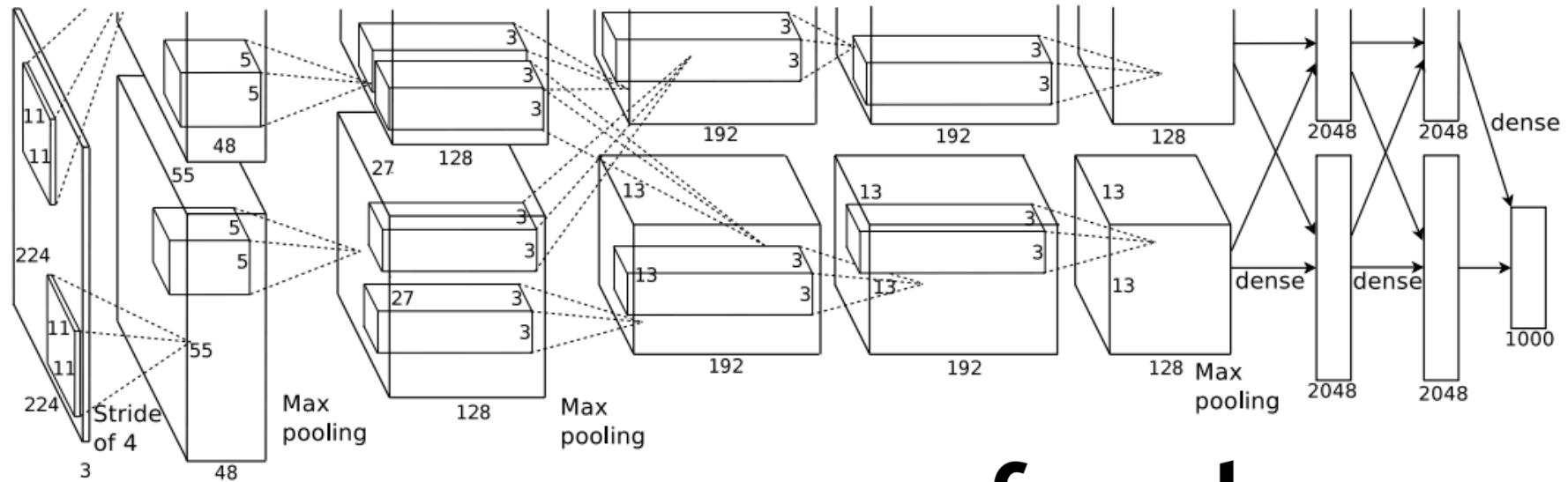
# AlexNet - 2012

- Input Image 227x227x3
- First Layer: 96 11x11 Conv filters stride 4 pad 0
- How Many Parameters?  $11 \times 11 \times 3 \times 96 = 34,848$



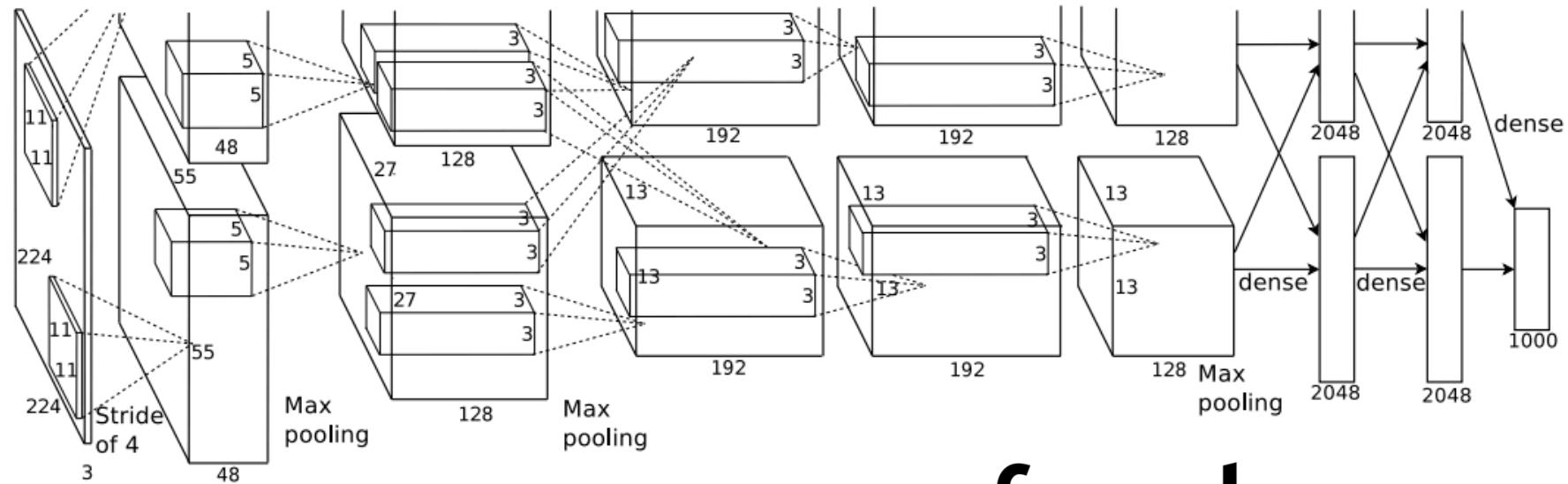
# AlexNet - 2012

- First Layer Output 55x55x96
- Second Layer: 2x2 Max Pooling stride 2
- What is the output size?



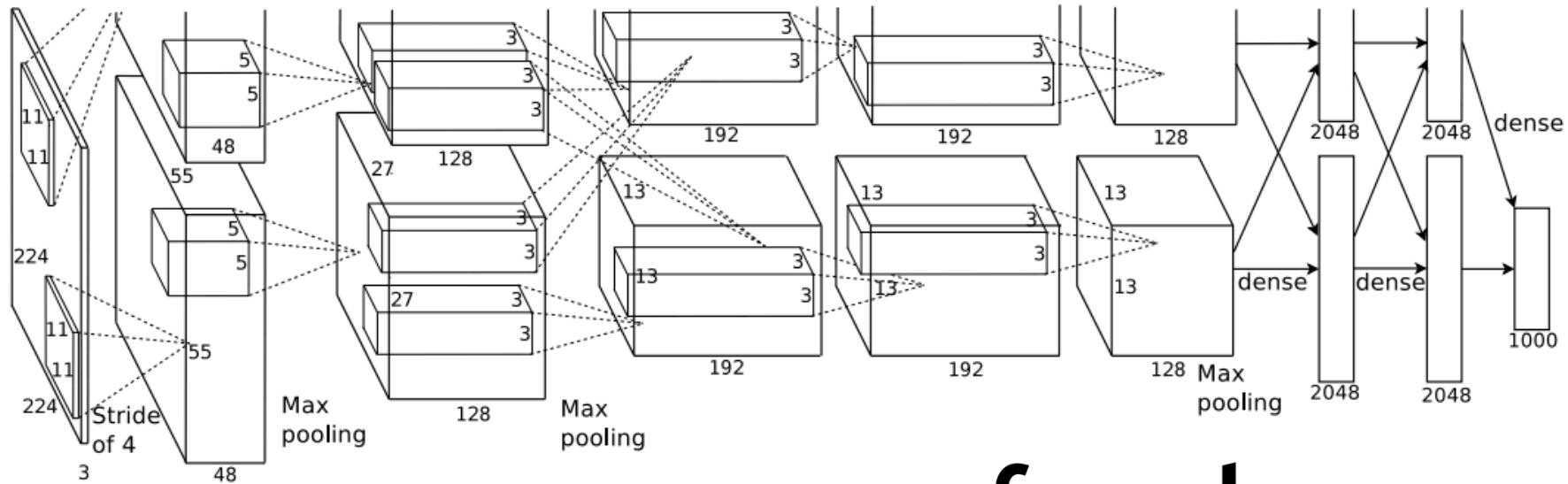
# AlexNet - 2012

- First Layer Output 55x55x96
- Second Layer: 2x2 Max Pooling stride 2
- What is the output size? 27x27x96



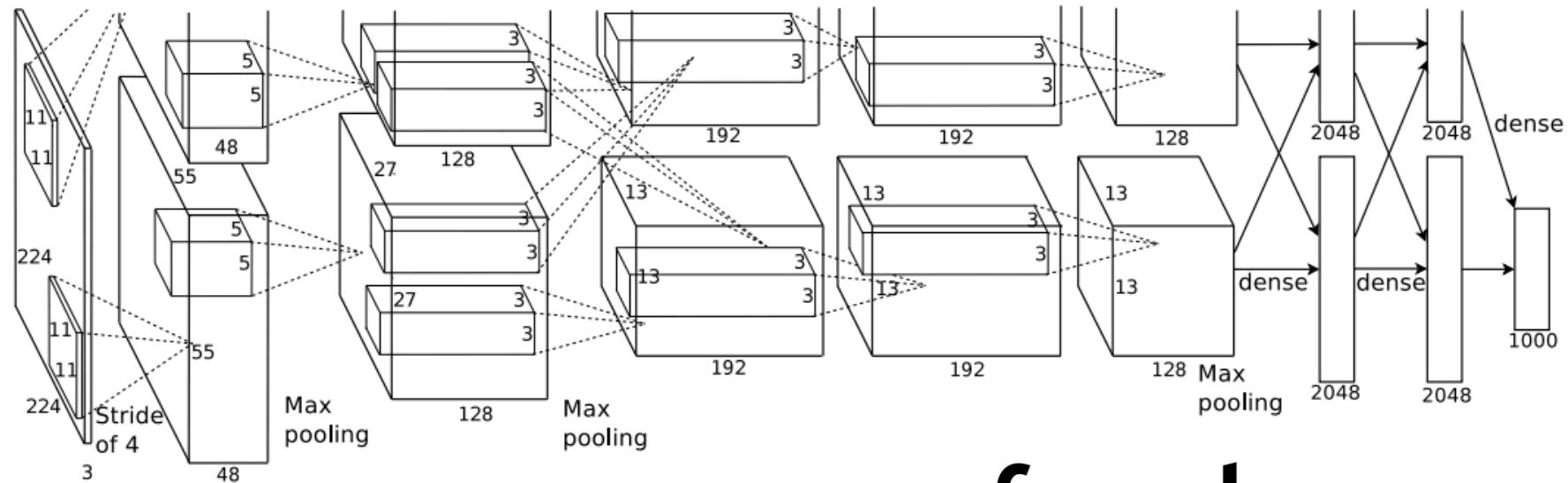
# AlexNet - 2012

- First Layer Output 55x55x96
- Second Layer: 2x2 Max Pooling stride 2
- What is the number of params?



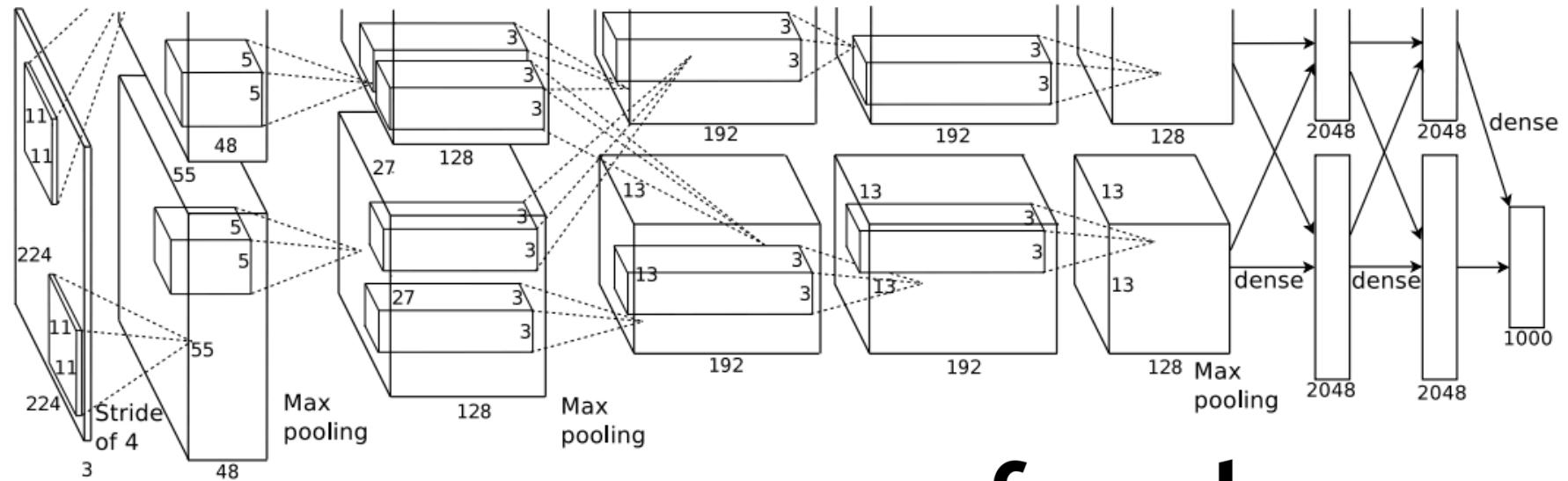
# AlexNet - 2012

- First Layer Output 55x55x96
- Second Layer: 2x2 Max Pooling stride 2
- What is the number of params? 0



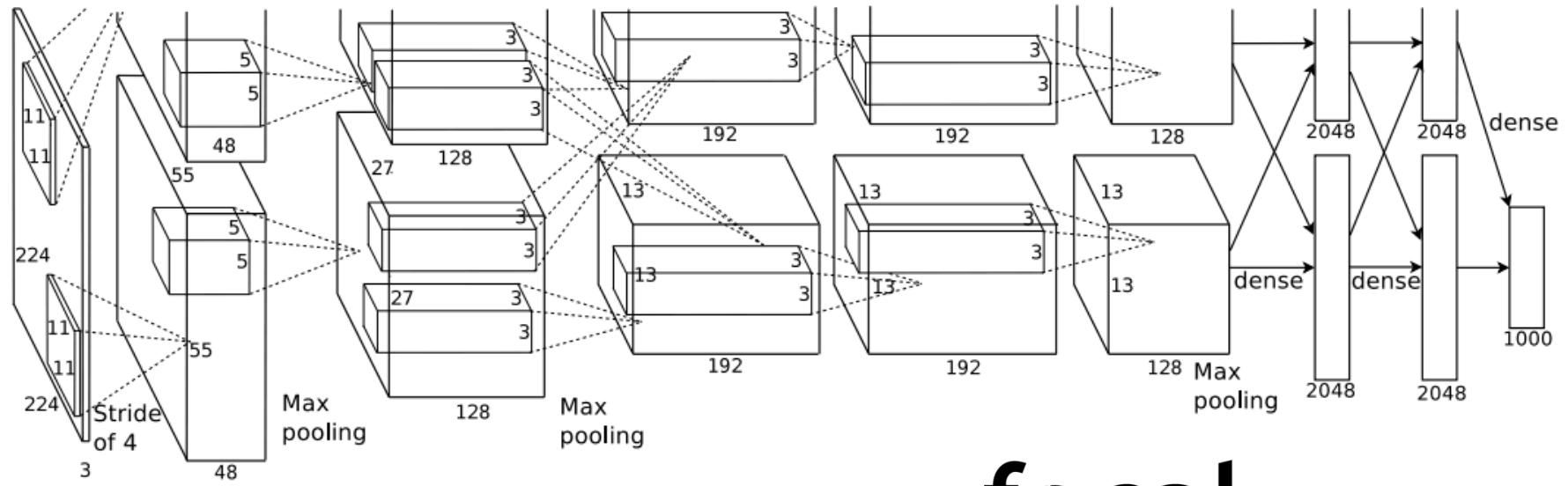
# AlexNet - 2012

- First use of dropout, relu
- Heavy data augmentation
- SGD + Momentum, lr=1e-2, momentum=0.9, l2=5e-4



# AlexNet - 2012

- GPUs too small, so they had half the network on different GPUs



# VGGNet 2014

- First simple widely used net
- Smaller and Deeper



AlexNet

VGG16



# VGGNet 2014

- First simple widely used net
- Smaller and Deeper

What is receptive field of 7x7 conv layer?



AlexNet

VGG16



# VGGNet 2014

- First simple widely used net
- Smaller and Deeper

What is receptive field of 7x7 conv layer?  
7x7



AlexNet

VGG16



# VGGNet 2014

- First simple widely used net
- Smaller and Deeper

What is receptive field of three  
3x3 conv layers?



AlexNet

VGG16



# VGGNet 2014

- First simple widely used net
- Smaller and Deeper

What is receptive field of three  
3x3 conv layers?  
Also 7x7



AlexNet

VGG16



# VGGNet 2014

- First simple widely used net
- Smaller and Deeper

How many params does a  $7 \times 7$  layer with depth  $D$  have vs three  $3 \times 3$  filters of depth  $D$ ?



AlexNet

VGG16

# VGGNet 2014

- First simple widely used net
- Smaller and Deeper

How many params does a 7x7 layer with depth D have vs three 3x3 filters of depth D?

$$7 \times 7 \times D \times D = 49D^2$$

$$3 \times (3 \times 3 \times D \times D) = 27D^2$$



AlexNet

VGG16



# VGGNet 2014

- First simple widely used net
- Smaller and Deeper

How many non-linearities in a 7x7 filter vs three 3x3 filters?



AlexNet

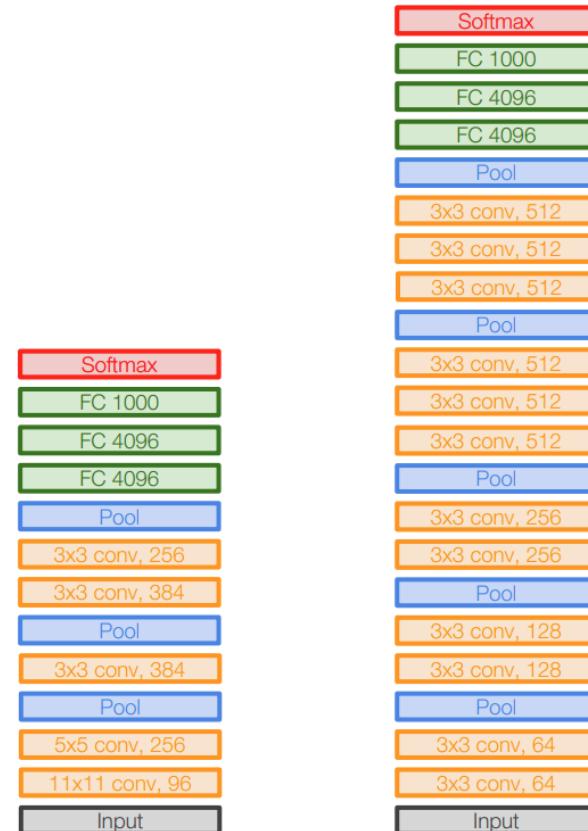
VGG16



# VGGNet 2014

- First simple widely used net
- Smaller and Deeper

How many non-linearities in a  
7x7 filter vs three 3x3 filters?  
1 vs 3



AlexNet

VGG16



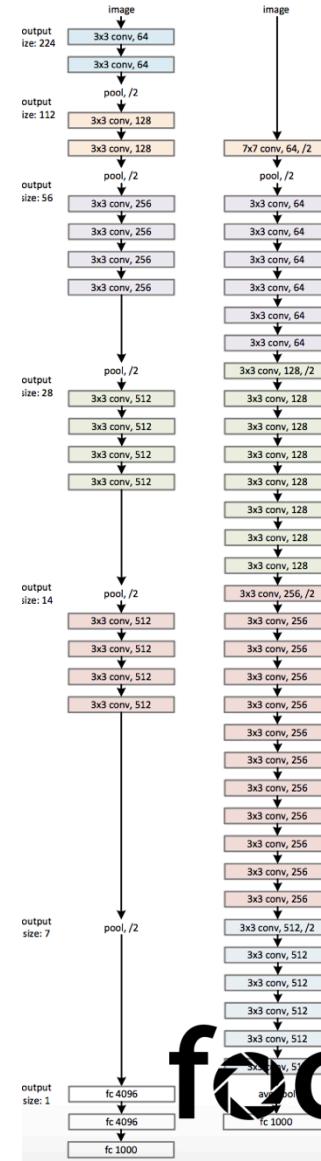
# VGGNet 2014

- Similar training procedure
- Good out of box model + weights
- Downsamples input by 32 and then does fully connected layer



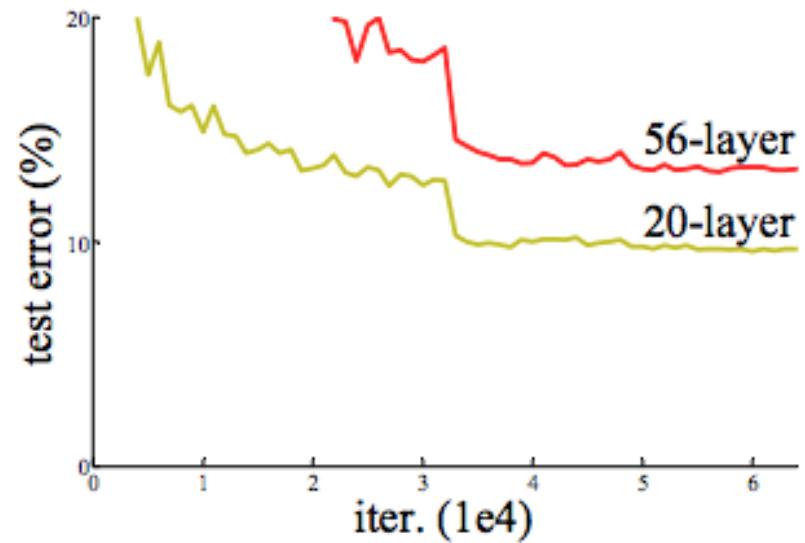
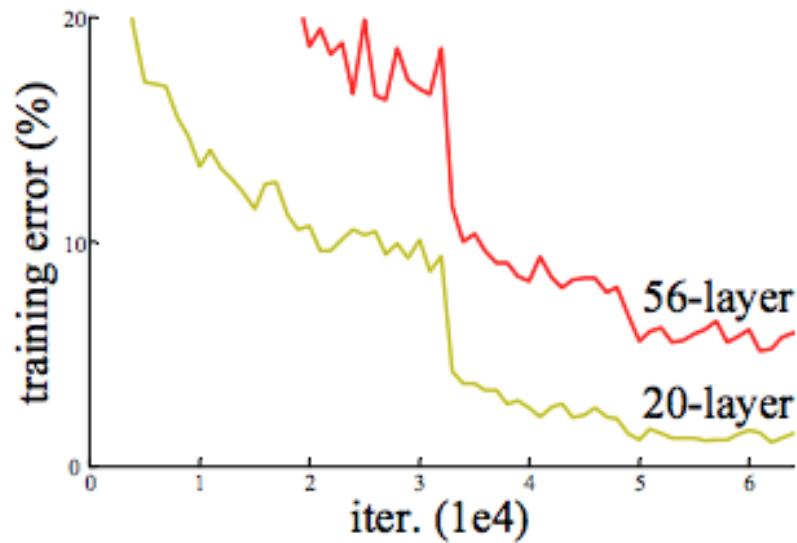
# ResNet 2015

- Lets go much deeper!
- Why not just add many convs?



# ResNet 2015

- Training curves for plain models



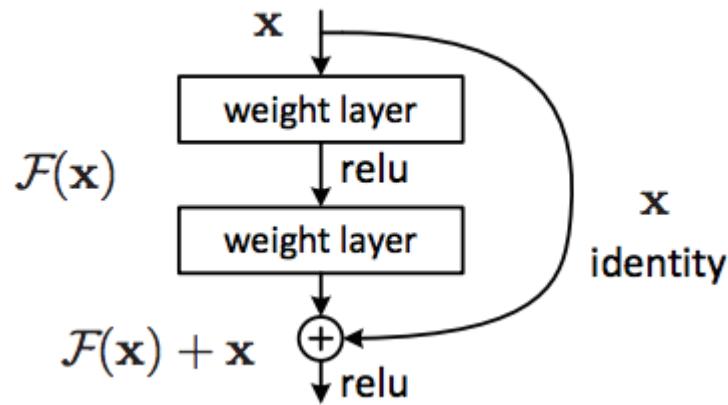
# ResNet 2015

- Training deep networks is a hard optimization problem

# ResNet 2015

- Solution: Identity Mappings

$$\mathcal{F}(\mathbf{x}) := \mathcal{H}(\mathbf{x}) - \mathbf{x}$$



# ResNet 2015

- Normal Networks:  $y = f(g(h(x)))$
- Residual Networks:  $y = f(g(h(x) + x)) + h(x) + g(h(x) + x) + h(x) + x$
- There is always direct gradient flow to  $x$

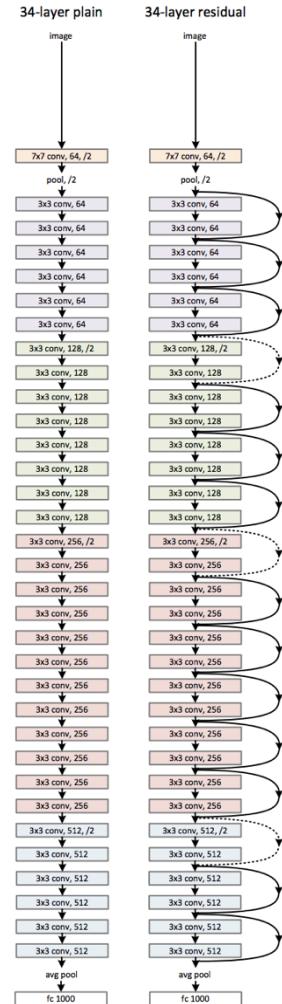
# ResNet 2015

- Showed continual improvement with increased depth

model	top-1 err.	top-5 err.
VGG-16 [41]	28.07	9.33
GoogLeNet [44]	-	9.15
PReLU-net [13]	24.27	7.38
plain-34	28.54	10.02
ResNet-34 A	25.03	7.76
ResNet-34 B	24.52	7.46
ResNet-34 C	24.19	7.40
ResNet-50	22.85	6.71
ResNet-101	21.75	6.05
ResNet-152	<b>21.43</b>	<b>5.71</b>

# ResNet 2015

- Great default
- final layer does global average pooling
- start with  $7 \times 7 \times 64$  conv followed by  $3 \times 3$  max pool stride 2.  $\leq$  standard input processing
- Uses Conv - Batch Norm - Relu
- No Dropout, No other pooling
- Uses  $1 \times 1$  Convs to downsize, and then  $3 \times 3$ s



# Architectures Summary

- Deeper is better
- Focus on gradient flow
- Conv - Batch Norm - ReLu
- Compositions of small filters are key: 3x3, 1x1
- Pretrained ImageNet weights are very good
- Keras:  
<https://github.com/fchollet/keras/tree/master/keras/applications>



Deep Learning for Retail