

CS181: Computer Vision

Popular CNNs (part#1)

(LeNet, AlexNet, VGGNet)

Dissecting CNNs

Tuesday, September 15th, 2026



Md Alimoor Reza

Associate Professor of Computer Science

Review: Last Lecture

- **Convolutional Neural Network (CNN):** good for computer vision (CV) tasks
 - Convolution operation
 - Nonlinearity (may or may not include this after a convolution)
 - Pooling operation (may or may not include this after a convolution)
 - CNN: convolutional layer + nonlinearity + pooling layer

What type of recognition problem?



Object Recognition

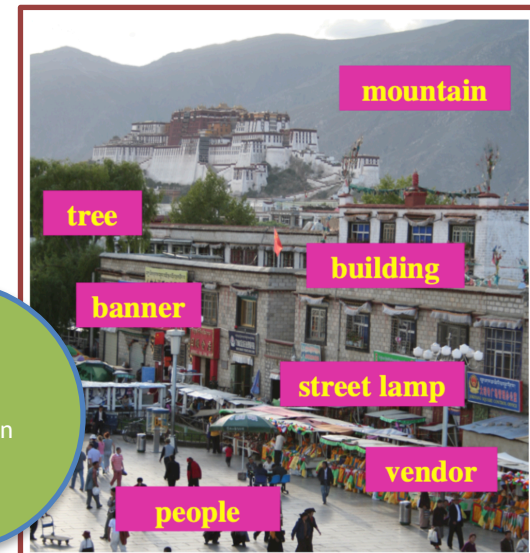


What type of recognition problem?



Single label classification

What type of recognition problem?



Multi-label classification

What type of recognition problem?



Scene and context classification

Topics

- Popular CNNs
 - LeNet
 - AlexNet
 - VGG
 - ResNet

Popular CNN: LeNet

- LeNet is a simple CNN architecture suitable for well-structured image
 - e.g., 28x28 pixels image of digits from 0 to 9 in MNIST or our Fashion-MNIST dataset

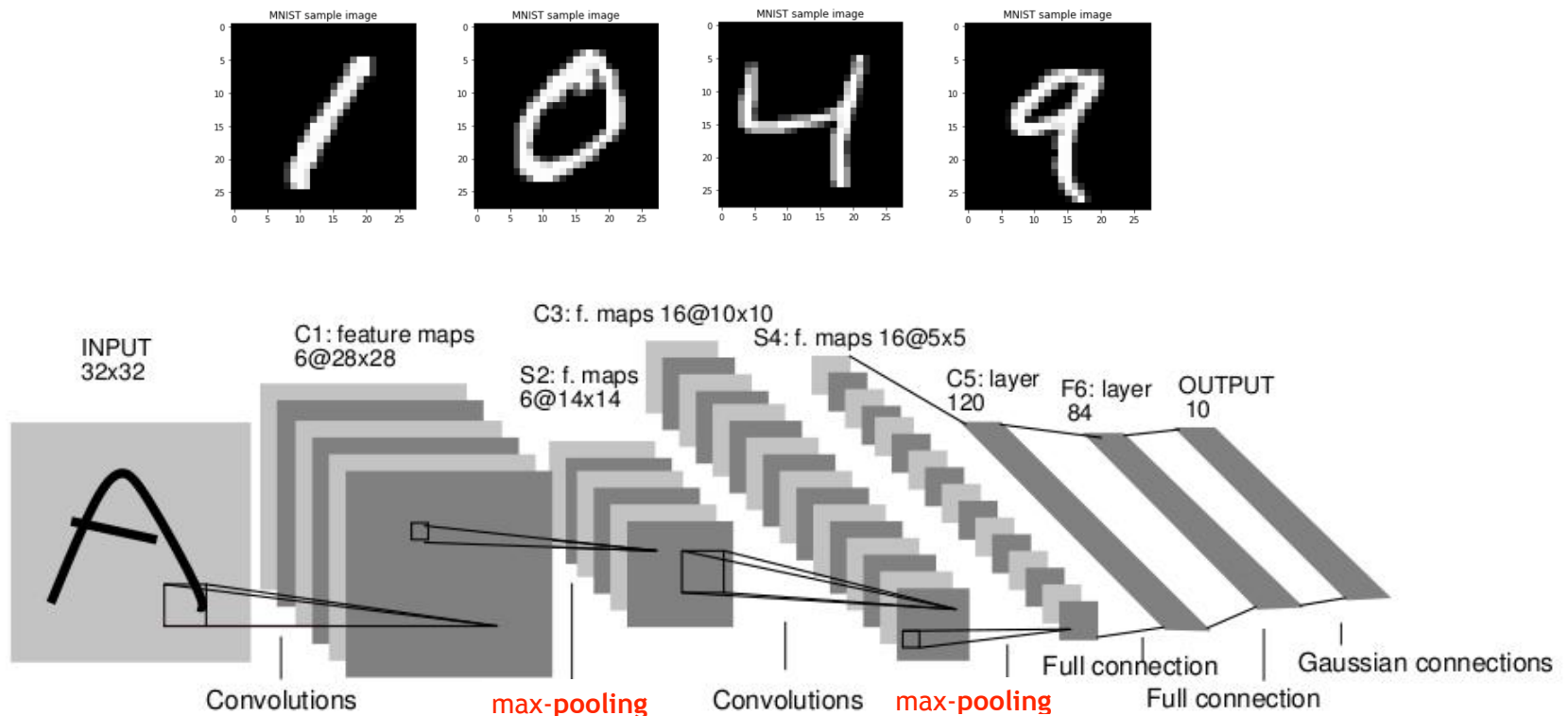


Figure: LeNet architecture as shown in the original paper. Here a pooling layer is also called a **subsampling** layer

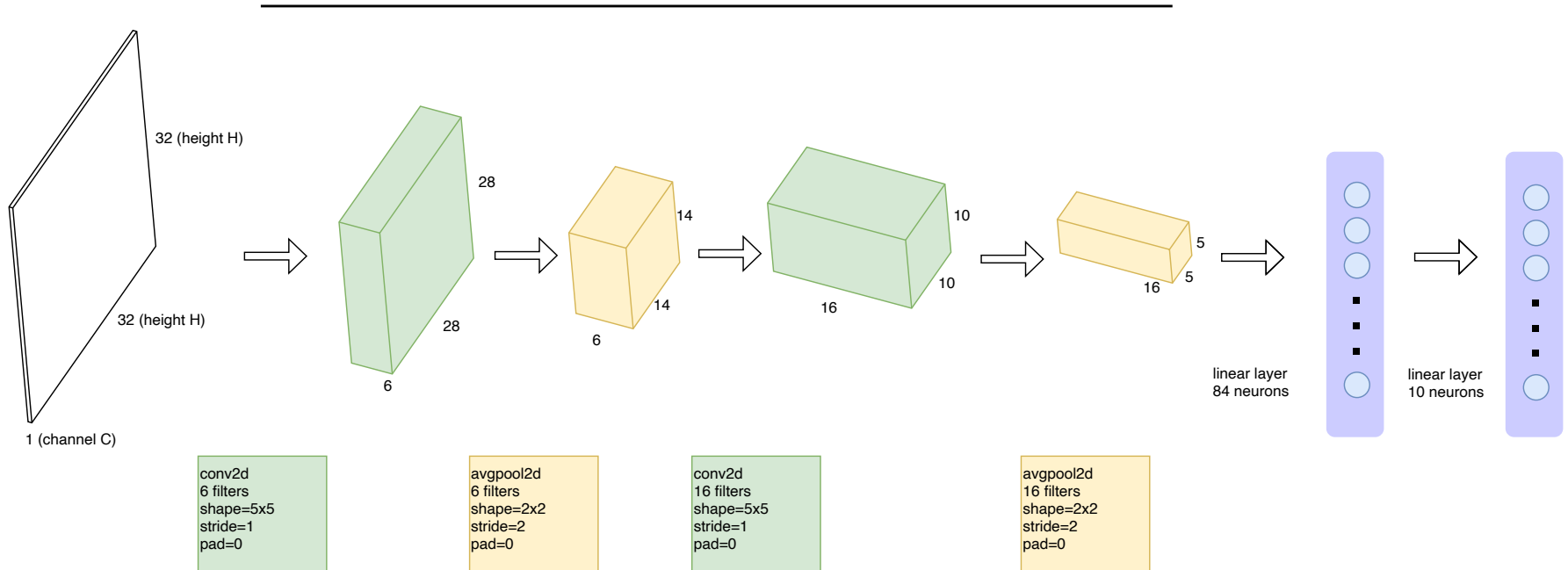
Group Activity

LeNet implementation

- Go to Blackboard and let's work on the notebook titled “CNN with activation and pooling layers”. Link shown below:

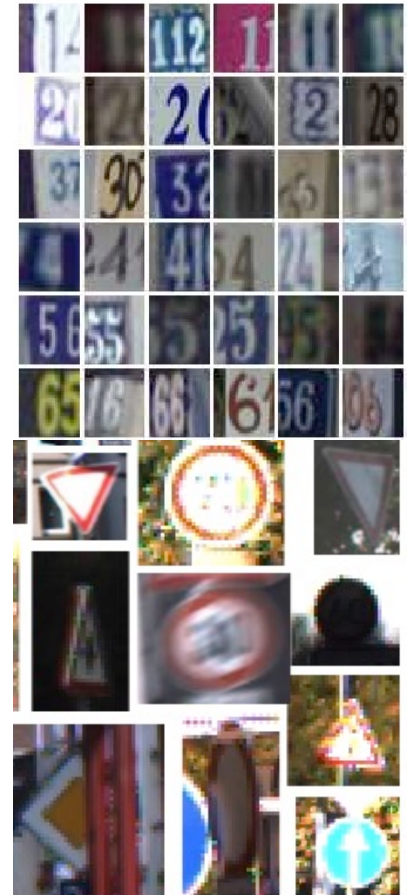
https://github.com/alimoorreza/CS181-fall26-notes/blob/main/day07_popular_cnn1_lenet_implementation.ipynb

LeNet Architecture



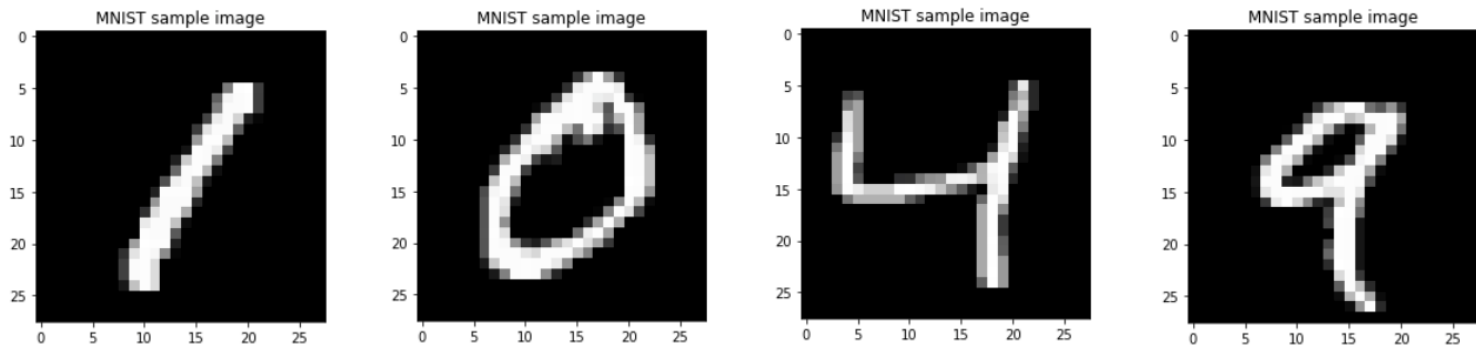
LeNet's success on simpler datasets

- Handwritten text/digits
 - MNIST (0.17% error [Ciresan et al. 2011])
 - Arabic and Chinese [Ciresan et al. 2012]
- Simpler recognition benchmarks
 - CIFAR-10 (9.3% error [Wan et al. 2013])
 - Traffic sign recognition
- Until 2011, it was less good at more complex datasets
 - Caltech-101/256 (few training examples)



LeNet insufficient for real-world images

- LeNet is a simple CNN architecture suitable for well-structured image
 - e.g., 28x28 pixels image of digits from 0 to 9 in MNIST or our Fashion-MNIST dataset



- Real-world images are much more complicated; pose challenges in classification
 - e.g., high resolution images 600x480 pixels image and contents have a lot more diversity



ImageNet Challenge

- **ImageNet dataset**

- 14 million labeled images
- 20k classes

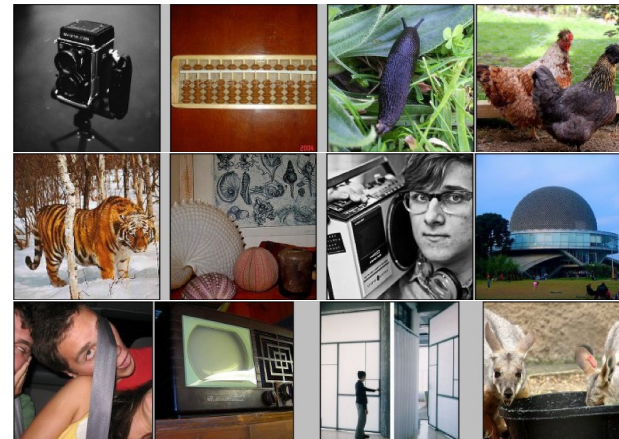


- **Data source**

- Images gathered from Internet

- **Image annotators**

- Human labels via Amazon Turk



Project led by Fei-Fei Li at Stanford
CVPR 2009

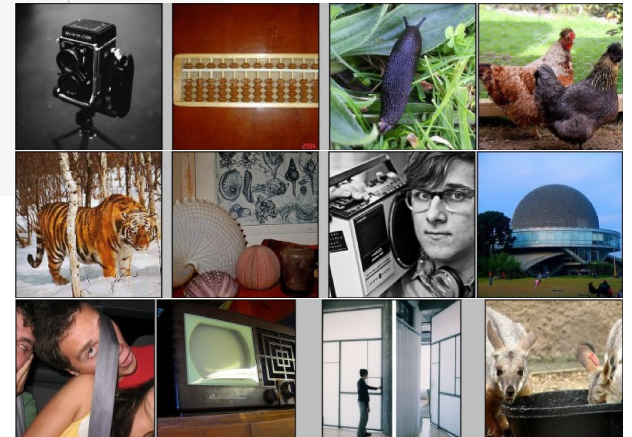
- **What is ImageNet challenge**

- Train your network with [a subset of 1.2 million training images](#) from ImageNet
- Test your network using another testing subset where you need to [classify each image to one of the 1000 classes](#)

ImageNet Challenge

```
imagenet_1000_classes_label_map = \n{\n0: 'tench, Tinca tinca',\n1: 'goldfish, Carassius auratus',\n2: 'great white shark, white shark, man-eater, man-eating shark, Carcharodon carcharias',\n3: 'tiger shark, Galeocerdo cuvieri',\n4: 'hammerhead, hammerhead shark',\n5: 'electric ray, crampfish, numbfish, torpedo',\n6: 'stingray',\n7: 'cock',\n8: 'hen',\n9: 'ostrich, Struthio camelus',\n10: 'brambling, Fringilla montifringilla',\n11: 'goldfinch, Carduelis carduelis',\n12: 'house finch, linnet, Carpodacus mexicanus',\n13: 'junco, snowbird',\n14: 'indigo bunting, indigo finch, indigo bird, Passerina cyanea',\n15: 'robin, American robin, Turdus migratorius',\n16: 'bulbul',\n17: 'jay',\n18: 'magpie',\n19: 'chickadee',\n20: 'water ouzel, dipper',
```

IMAGENET



Project led by Fei-Fei Li at Stanford
CVPR 2009

ImageNet Challenge 2012

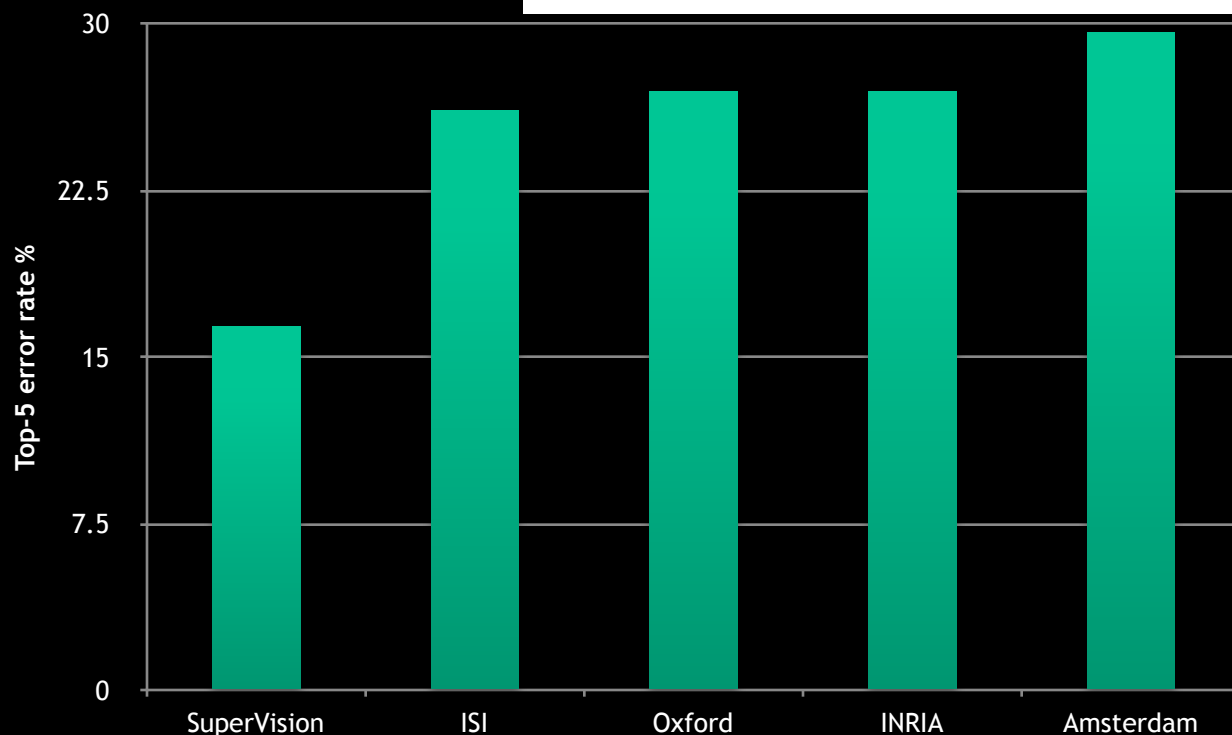


NeurIPS 2016

<https://papers.nips.cc/paper/4824-Imagenet-classification>

ImageNet Classification with Deep Convolutional Neural ...

by A Krizhevsky · 2012 · Cited by 158630 — We trained a large, **deep convolutional neural network** to classify the 1.3 million high-resolution images in the LSVRC-2010 **ImageNet** training set into the 1000 ...



- AlexNet (Krizhevsky et al.) -- **16.4% error** (top-5)
- Next best (non-convnet) – **26.2% error**

Topics

- Popular CNNs
 - LeNet
 - AlexNet
 - VGG
 - ResNet

Popular CNN: AlexNet

- AlexNet's important features. Similar framework to LeNet with important distinction such as:

- Bigger model compared to LeNet
 - 5 convolution layers + 3 linear layers
 - 60,000,000 params
- Trained on more data
 - 10^6 images (ImageNet) vs. 10^3 images (digit images)
- GPU implementation (50x speedup over CPU)
 - trained on two GPUs for a week
- Better regularization for training
 - introduced a new technique called **Dropout**
 - **Dropout**: randomly turning off neurons from the network

ImageNet Classification with Deep Convolutional Neural Networks

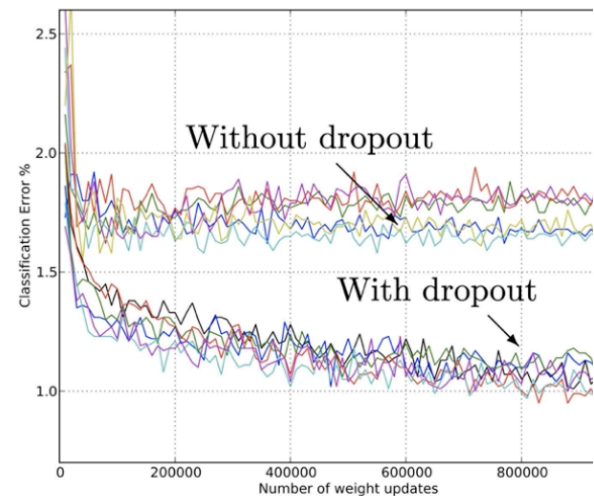
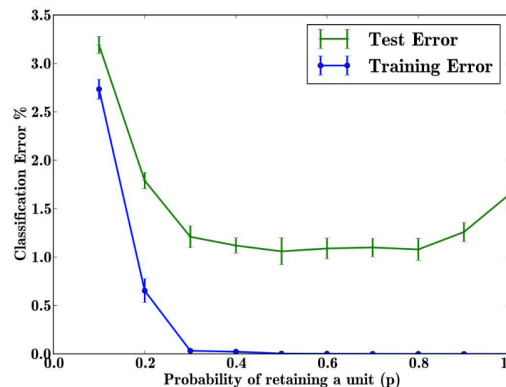
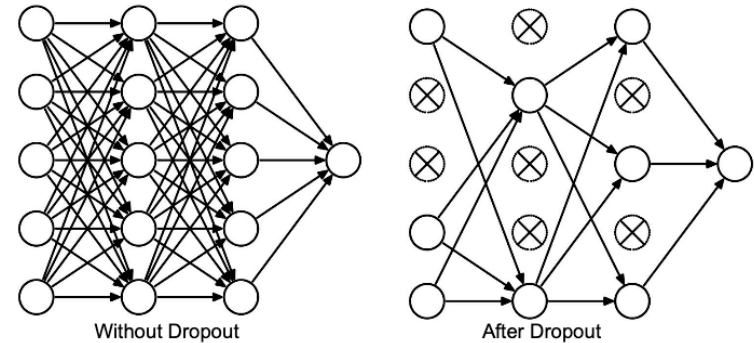
Alex Krizhevsky Ilya Sutskever Geoffrey E. Hinton
University of Toronto University of Toronto University of Toronto
kriz@cs.utoronto.ca ilya@cs.utoronto.ca hinton@cs.utoronto.ca

Abstract

We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet ILSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called "dropout" that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.

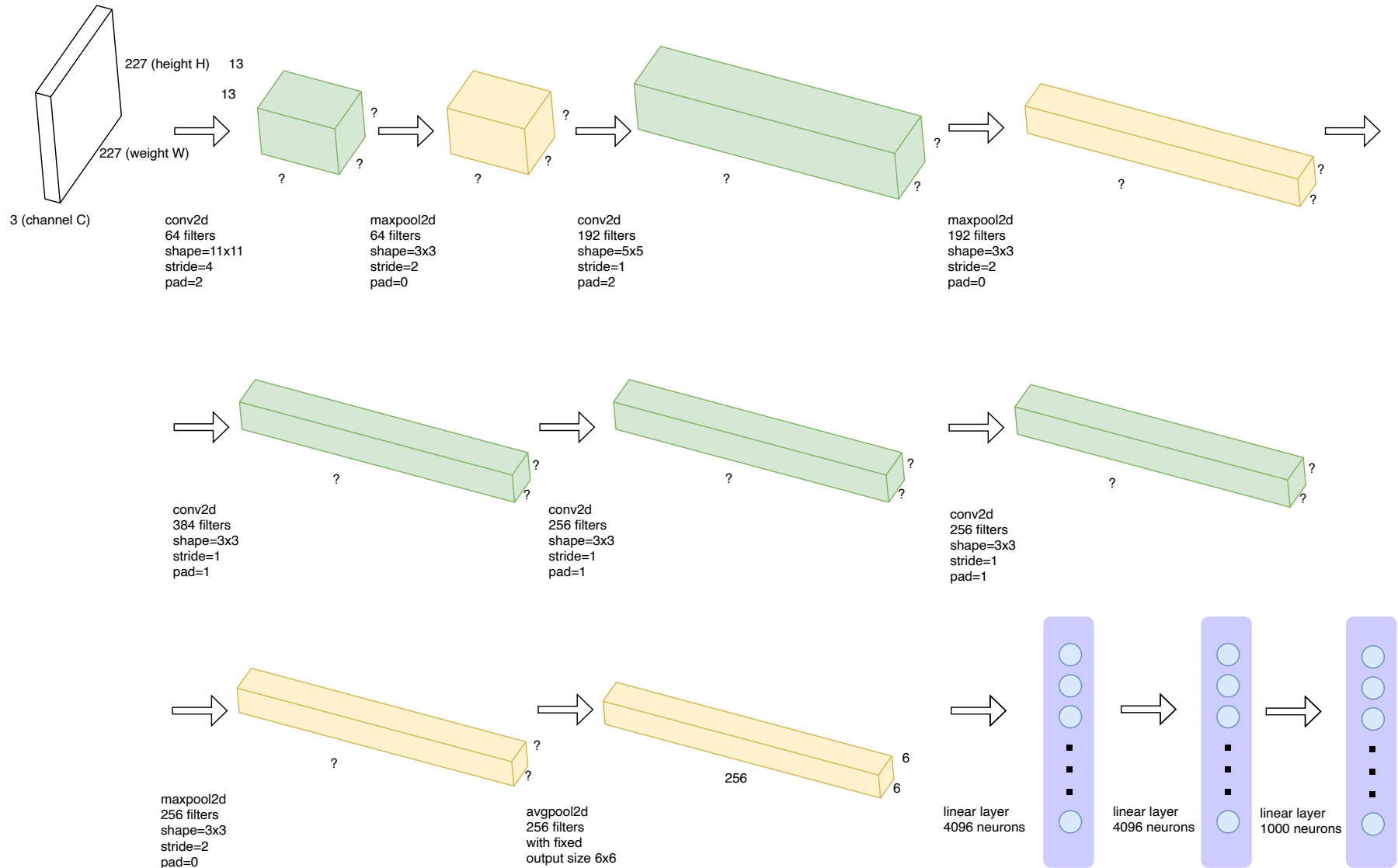
Dropout

- AlexNet's important features:
 - Better regularization during training a neural network:
 - introduced a new technique called **Dropout**
 - **Dropout**: randomly set output of a hidden unit to **zero** with a probability p
 - **Dropped out unit** has no contribution in forward of backward passes
- Choosing dropout rate:
 - p : probability of retaining a unit (range 0.5-0.8)
 - n : number of hidden units
 - smaller p leads to under-fitting
 - large p may lead to overfitting



Group Activity

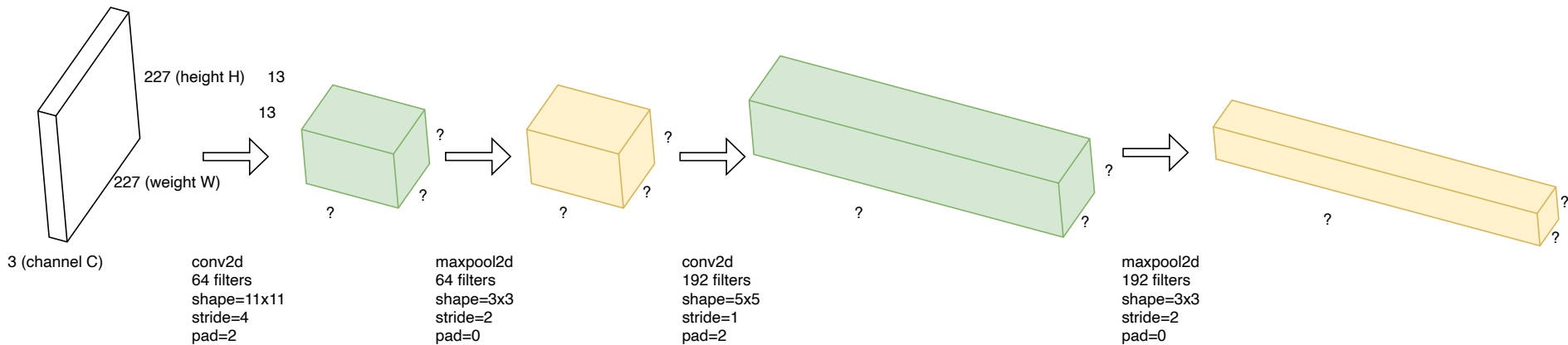
What are the dimensions of different volumes?



A. Krizhevsky, I. Sutskever, and G. Hinton, [ImageNet Classification with Deep Convolutional Neural Networks](#), NIPS 2012

Solution: Group Activity

What are the dimensions of different volumes?



Convolution output volume
($C_2 \times H_2 \times W_2$)

$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$$

$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1$$

$$C_2 = K \text{ ***number of filters***}$$

Pooling output volume
($C_2 \times H_2 \times W_2$)

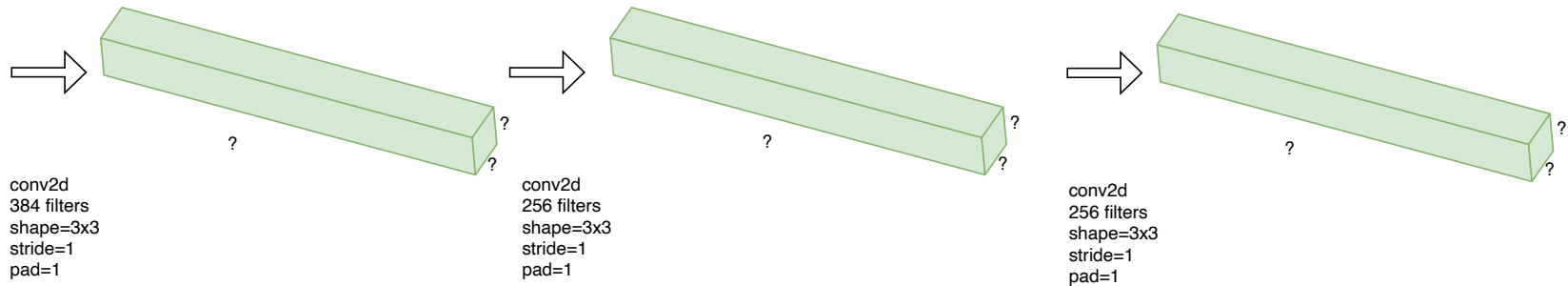
$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$$

$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1$$

$$C_2 = C_1 \text{ ***input volume's channel***}$$

Solution: Group Activity

What are the dimensions of different volumes?



Convolution output volume
($C_2 \times H_2 \times W_2$)

$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$$

$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1$$

$$C_2 = K \text{ ***number of filters***}$$

Pooling output volume
($C_2 \times H_2 \times W_2$)

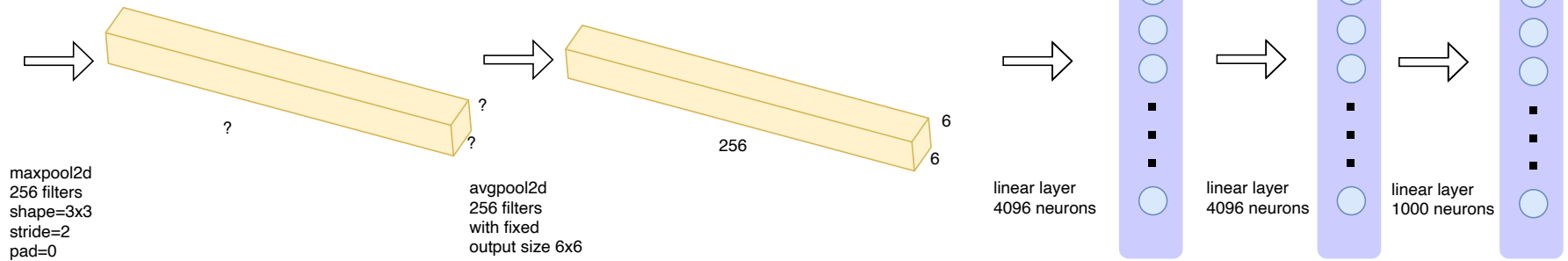
$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$$

$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1$$

$$C_2 = C_1 \text{ ***input volume's channel***}$$

Solution: Group Activity

What are the dimensions of different volumes?



Convolution output volume
($C_2 \times H_2 \times W_2$)

$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$$

$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1$$

$$C_2 = K \text{ ***number of filters***}$$

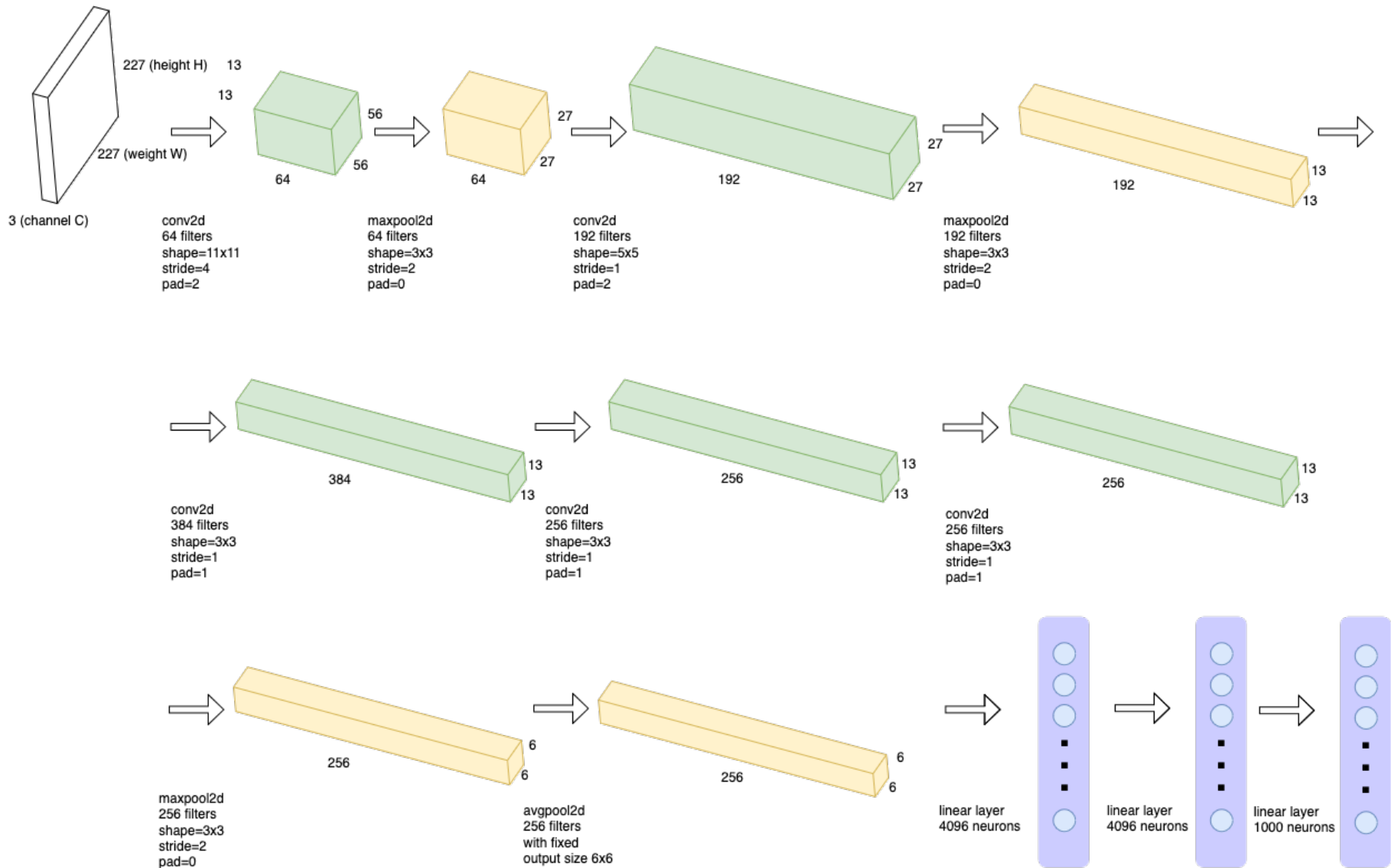
Pooling output volume
($C_2 \times H_2 \times W_2$)

$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$$

$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1$$

$$C_2 = C_1 \text{ ***input volume's channel***}$$

Full solution: the dimensions of different volumes



A. Krizhevsky, I. Sutskever, and G. Hinton, [ImageNet Classification with Deep Convolutional Neural Networks](#), NIPS 2012

Popular CNN: AlexNet

```
[torch.Size([3, 227, 227])] INPUT
[torch.Size([64, 56, 56])] CONV1: 64 convolutional filters each with a shape=11x11, stride=4, pad=2
[torch.Size([64, 27, 27])] MAX POOL1: 3x3 max pooling filters with stride=2, pad=0
[torch.Size([192, 27, 27])] CONV2: 192 convolutional filters each with a shape=5x5, stride=1, pad=2
[torch.Size([192, 13, 13])] MAX POOL2: 3x3 max pooling filters with stride=2, pad=0
[torch.Size([384, 13, 13])] CONV3: 384 convolutional filters each with a shape=3x3, stride=1, pad=1
[torch.Size([256, 13, 13])] CONV4: 256 convolutional filters each with a shape=3x3, stride=1, pad=1
[torch.Size([256, 13, 13])] CONV5: 256 convolutional filters each with a shape=3x3, stride=1, pad=1
[torch.Size([256, 6, 6])] MAX POOL3: 3x3 max pooling filters with stride=2, pad=0
[torch.Size([256, 6, 6])] AVG POOL: fixed output size [width x height]
[torch.Size([4096])] FC6: neurons
[torch.Size([4096])] FC7: neurons
[torch.Size([4])] FC8: neurons
```

Topics

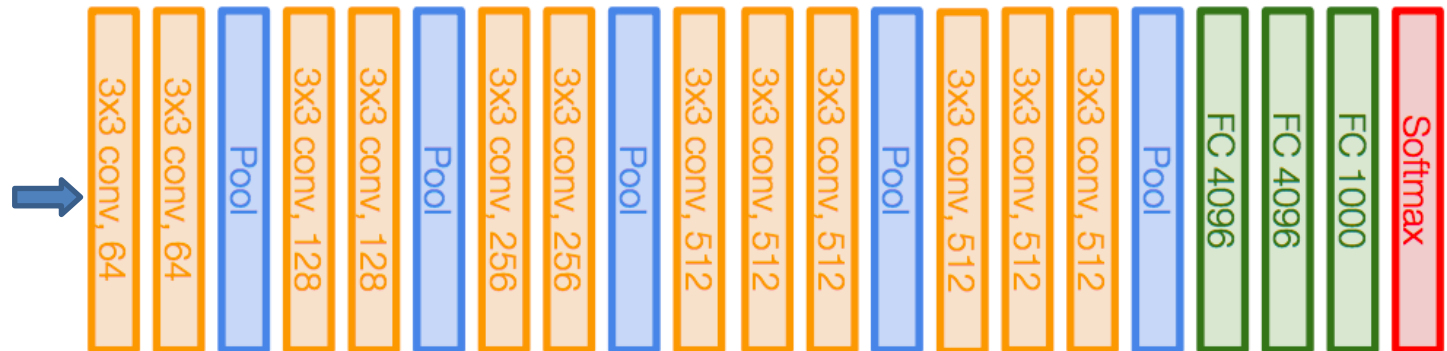
- Popular CNNs
 - LeNet
 - AlexNet
 - VGG
 - ResNet

Popular CNN: VGG-16

- VGG-16 was the winner of ImageNet (1000-class image classification) challenge in 2014
 - proposed by Oxford University (Andrew Zisserman's group)

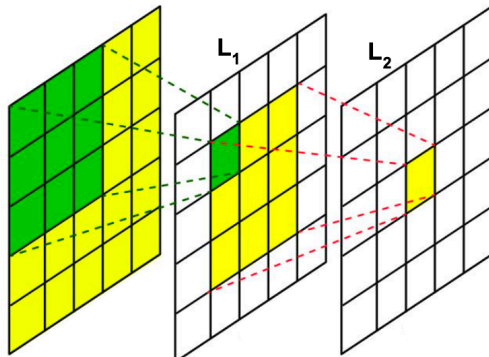


Input image



Popular CNN: VGG-16

- VGG-16's important features. VGG was the winner of ImageNet challenge in 2014:
- Bigger model compared to AlexNet
 - vgg16: 13 convolutional layers + 3 linear layers
 - vgg19: 16 convolutional layers + 3 linear layers
 - last 3 linear layers are the same as AlexNet
 - approximately 140,000,000 params
- Stacking 3x3 convolution filters:
 - fixed 3x3 convolution filters but many of them
 - within a sublayer, multiple 3x3 convolution filters are stacked together
 - stacking them together reduces the number of parameters needed to cover similar fields of view



Published as a conference paper at ICLR 2015

VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION

Karen Simonyan* & Andrew Zisserman*

Visual Geometry Group, Department of Engineering Science, University of Oxford
{karen,az}@robots.ox.ac.uk

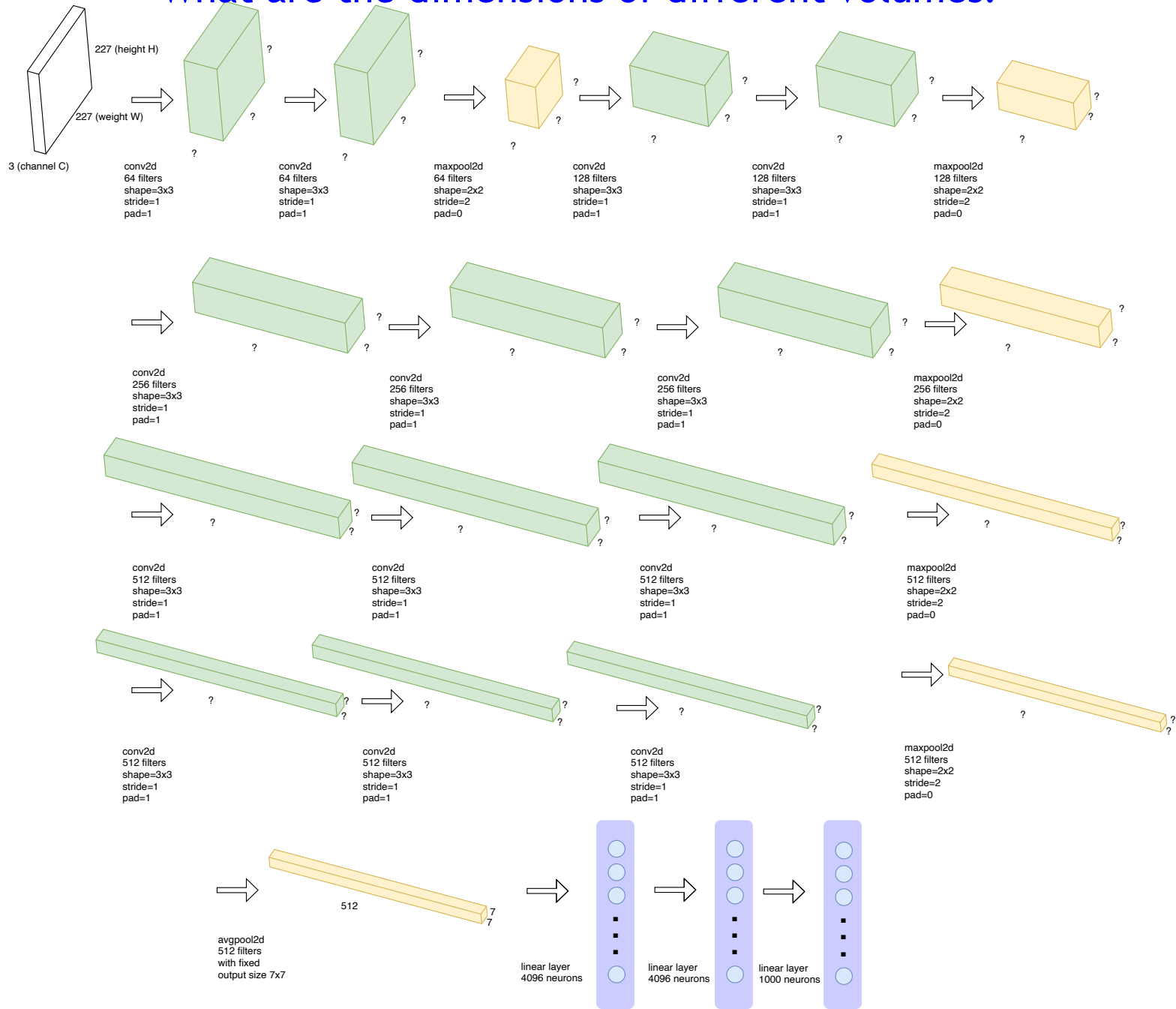
ABSTRACT

In this work we investigate the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. Our main contribution is a thorough evaluation of networks of increasing depth using an architecture with very small (3×3) convolution filters, which shows that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 16–19 weight layers. These findings were the basis of our ImageNet Challenge 2014 submission, where our team secured the first and the second places in the localisation and classification tracks respectively. We also show that our representations generalise well to other datasets, where they achieve state-of-the-art results. We have made our two best-performing ConvNet models publicly available to facilitate further research on the use of deep visual representations in computer vision.

Karen Simonyan and Andrew Zisserman, [Very Deep Convolutional Networks for Large-Scale Image Recognition](#), ICLR 2015

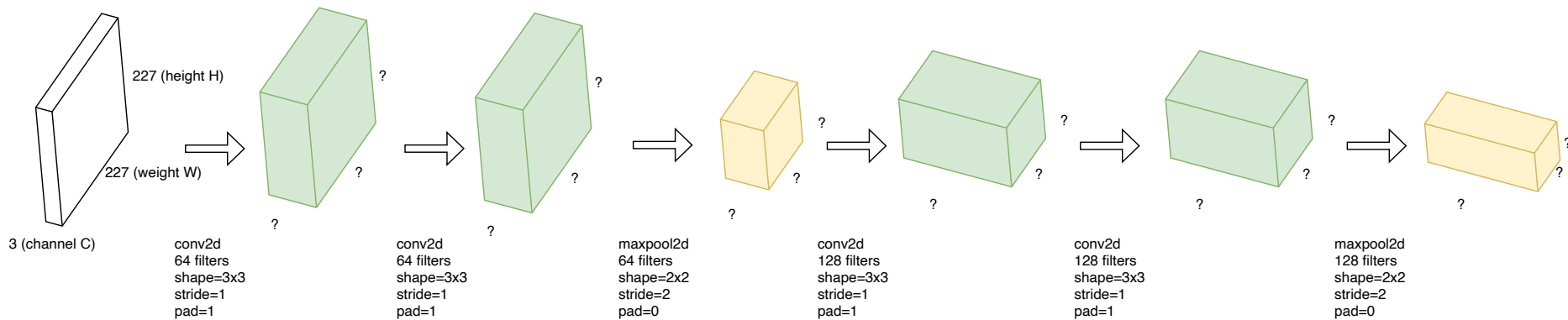
Group Activity

What are the dimensions of different volumes?



Group Activity

What are the dimensions of different volumes?



Convolution output volume
($C_2 \times H_2 \times W_2$)

$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$$

$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1$$

$$C_2 = K \text{ ***number of filters***}$$

Pooling output volume
($C_2 \times H_2 \times W_2$)

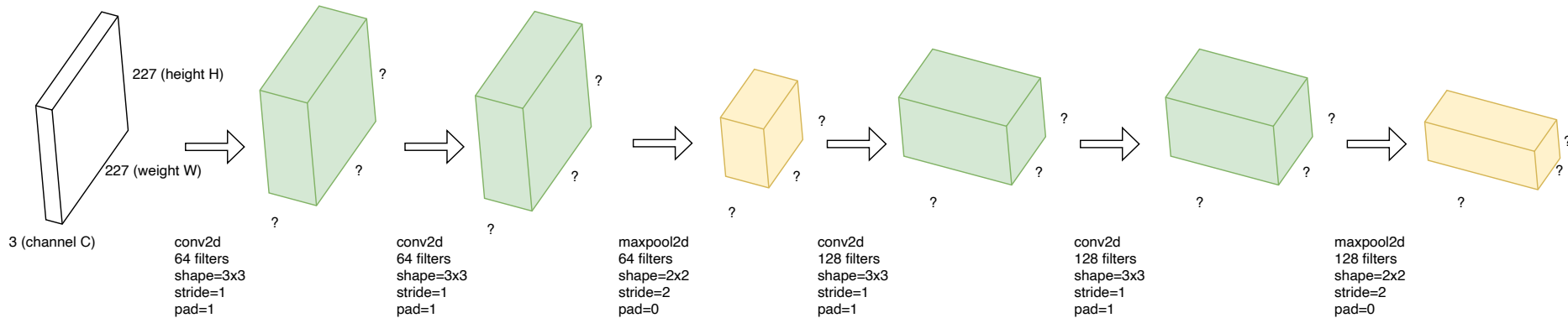
$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$$

$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1$$

$$C_2 = C_1 \text{ ***input volume's channel***}$$

Solution: Group Activity

What are the dimensions of different volumes?



Convolution output volume
($C_2 \times H_2 \times W_2$)

$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$$

$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1$$

$$C_2 = K \text{ ***number of filters***}$$

Pooling output volume
($C_2 \times H_2 \times W_2$)

$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$$

$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1$$

$$C_2 = C_1 \text{ ***input volume's channel***}$$

Conv#1

$$W = (227 - 3 + 2 \cdot 1) / 1 + 1 = 227$$

Conv#2

$$W = (227 - 3 + 2 \cdot 1) / 1 + 1 = 227$$

Pooling#1

$$W = (227 - 2 + 2 \cdot 0) / 2 + 1 = 113$$

Conv#3

$$W = (113 - 3 + 2 \cdot 1) / 1 + 1 = 113$$

Conv#4

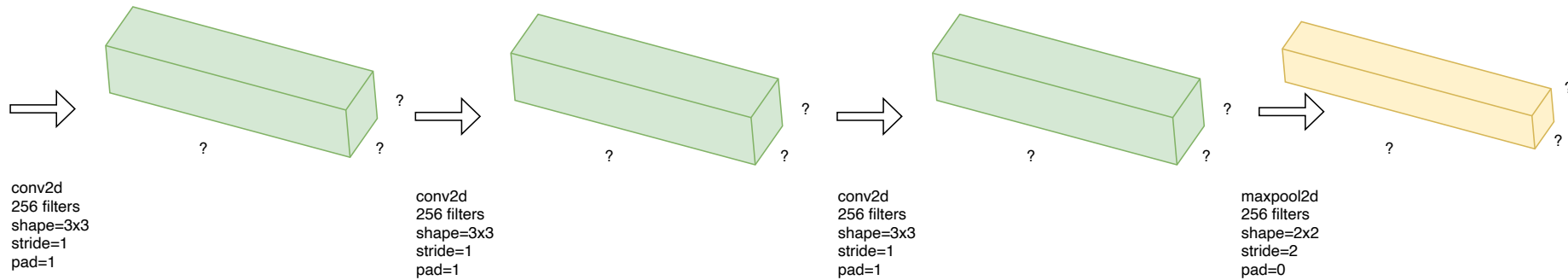
$$W = (113 - 3 + 2 \cdot 1) / 1 + 1 = 113$$

Pooling#2

$$W = (113 - 2 + 2 \cdot 0) / 2 + 1 = 56$$

Group Activity

What are the dimensions of different volumes?



Convolution output volume
($C_2 \times H_2 \times W_2$)

$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$$

$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1$$

$$C_2 = K \text{ ***number of filters***}$$

Pooling output volume
($C_2 \times H_2 \times W_2$)

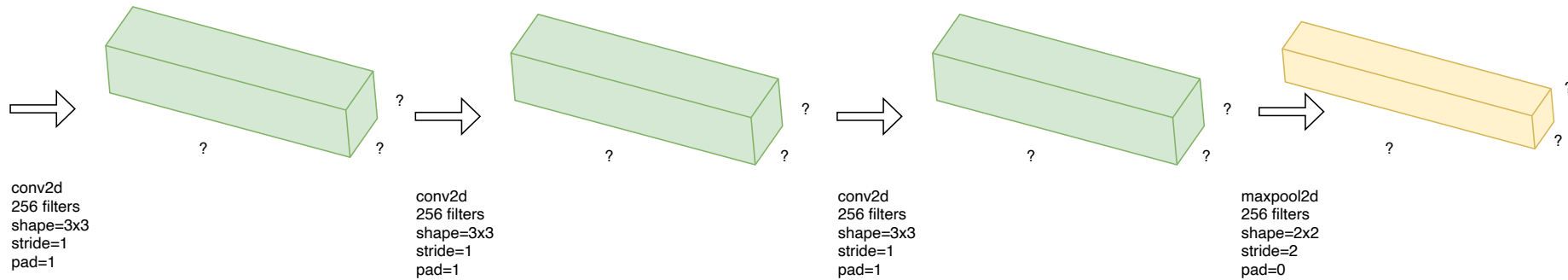
$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$$

$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1$$

$$C_2 = C_1 \text{ ***input volume's channel***}$$

Solution: Group Activity

What are the dimensions of different volumes?



Convolution output volume
($C_2 \times H_2 \times W_2$)

$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$$

$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1$$

$$C_2 = K \text{ ***number of filters***}$$

Pooling output volume
($C_2 \times H_2 \times W_2$)

$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$$

$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1$$

$$C_2 = C_1 \text{ ***input volume's channel***}$$

Conv#5

$$W = (56 - 3 + 2 \cdot 1) / 1 + 1 = 56$$

Conv#6

$$W = (56 - 3 + 2 \cdot 1) / 1 + 1 = 56$$

Conv#7

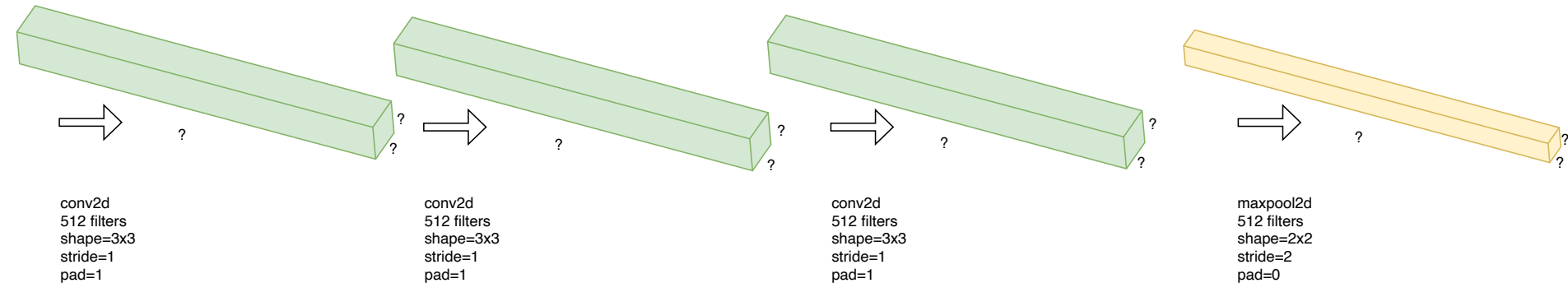
$$W = (56 - 3 + 2 \cdot 1) / 1 + 1 = 56$$

Pooling#3

$$W = (56 - 2 + 2 \cdot 0) / 2 + 1 = 28$$

Group Activity

What are the dimensions of different volumes?



Convolution output volume
($C_2 \times H_2 \times W_2$)

$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$$

$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1$$

$$C_2 = K \text{ ***number of filters***}$$

Pooling output volume
($C_2 \times H_2 \times W_2$)

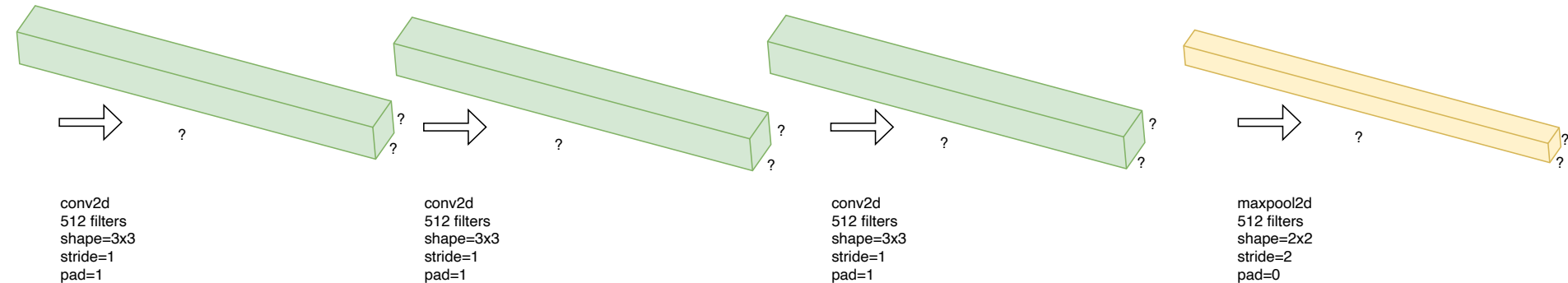
$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$$

$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1$$

$$C_2 = C_1 \text{ ***input volume's channel***}$$

Solution: Group Activity

What are the dimensions of different volumes?



Convolution output volume
($C_2 \times H_2 \times W_2$)

$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$$

$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1$$

$$C_2 = K \text{ ***number of filters***}$$

Pooling output volume
($C_2 \times H_2 \times W_2$)

$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$$

$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1$$

$$C_2 = C_1 \text{ ***input volume's channel***}$$

Conv#5

$$W = (28 - 3 + 2 \cdot 1) / 1 + 1 = 28$$

Conv#6

$$W = (28 - 3 + 2 \cdot 1) / 1 + 1 = 28$$

Conv#7

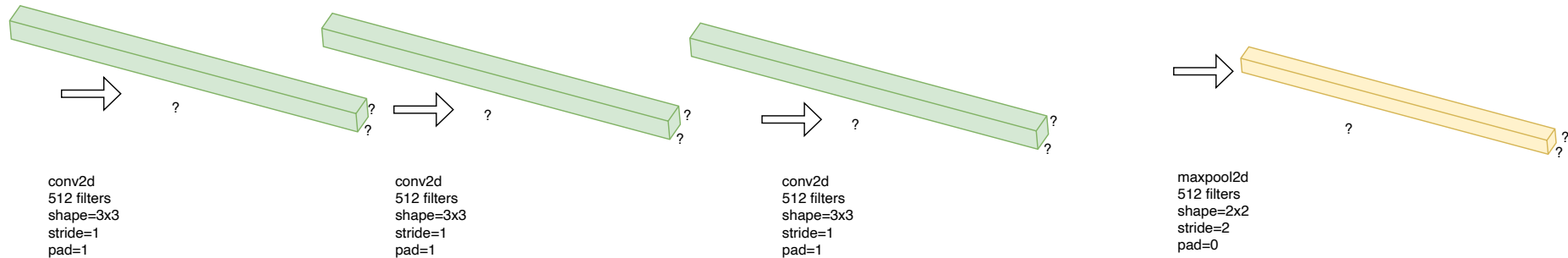
$$W = (28 - 3 + 2 \cdot 1) / 1 + 1 = 28$$

Pooling#3

$$W = (28 - 2 + 2 \cdot 0) / 2 + 1 = 14$$

Group Activity

What are the dimensions of different volumes?



Convolution output volume
($C_2 \times H_2 \times W_2$)

$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$$

$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1$$

$$C_2 = K \text{ ***number of filters***}$$

Pooling output volume
($C_2 \times H_2 \times W_2$)

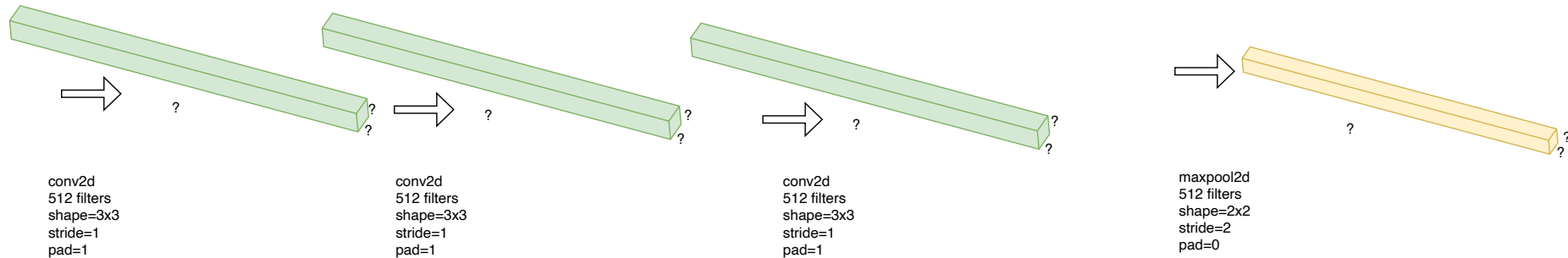
$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$$

$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1$$

$$C_2 = C_1 \text{ ***input volume's channel***}$$

Solution: Group Activity

What are the dimensions of different volumes?



Convolution output volume
($C_2 \times H_2 \times W_2$)

$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$$

$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1$$

$$C_2 = K \text{ ***number of filters***}$$

Pooling output volume
($C_2 \times H_2 \times W_2$)

$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$$

$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1$$

$$C_2 = C_1 \text{ ***input volume's channel***}$$

Conv#11

$$W = (14 - 3 + 2 \cdot 1) / 1 + 1 = 14$$

Conv#12

$$W = (14 - 3 + 2 \cdot 1) / 1 + 1 = 14$$

Conv#13

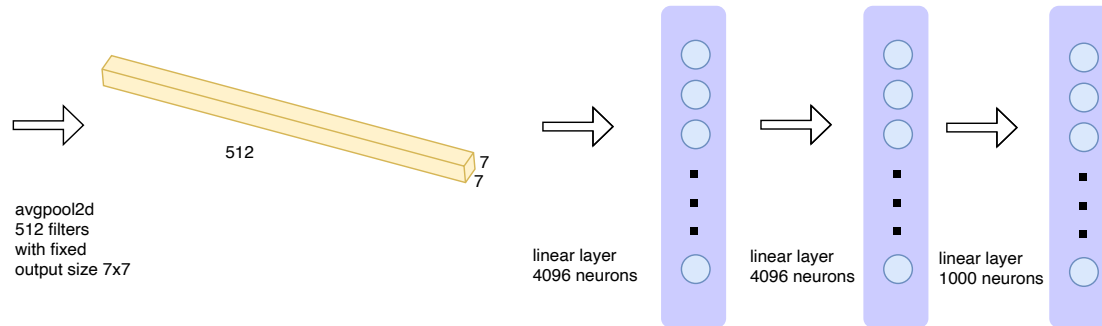
$$W = (14 - 3 + 2 \cdot 1) / 1 + 1 = 14$$

Pooling#5

$$W = (14 - 2 + 2 \cdot 0) / 2 + 1 = 7$$

Solution: Group Activity

What are the dimensions of different volumes?



Convolution output volume
($C_2 \times H_2 \times W_2$)

$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$$
$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1$$

$$C_2 = K \text{ ***number of filters***}$$

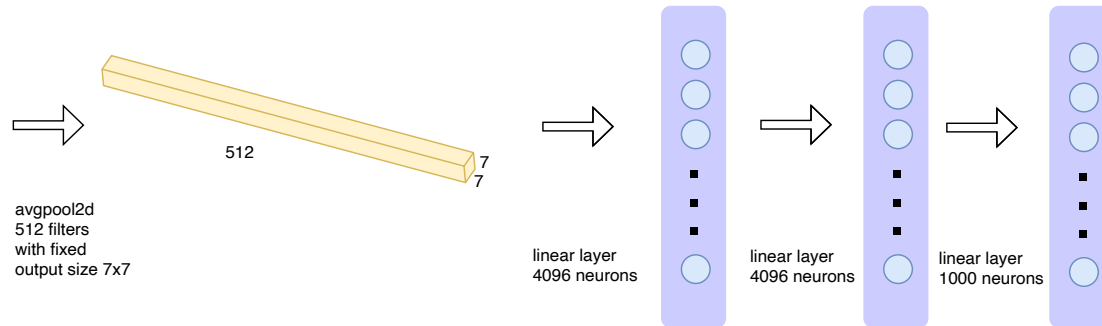
Pooling output volume
($C_2 \times H_2 \times W_2$)

$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1$$
$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1$$

$$C_2 = C_1 \text{ ***input volume's channel***}$$

Solution: Group Activity

What are the dimensions of different volumes?



Connections between pooling and linear layer

$$(C_2 \times H_2 \times W_2) \times 4096 \\ = (512 \times 7 \times 7) \times 4096$$

input volume's channel

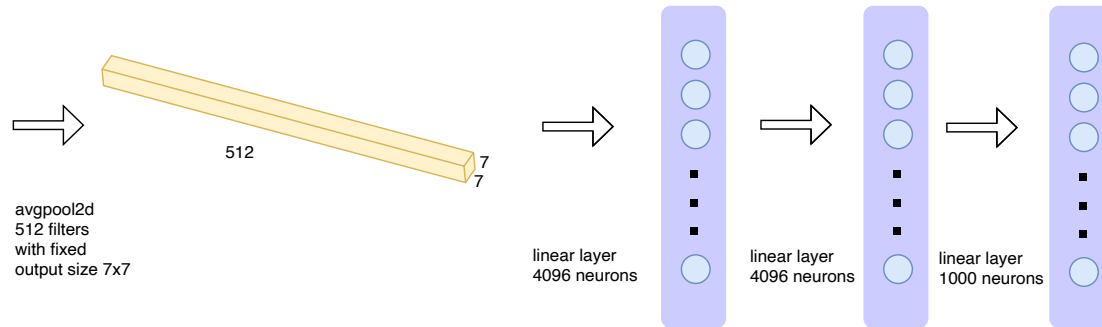
input volume's height

input volume's width

output number of neurons

Solution: Group Activity

What are the dimensions of different volumes?



Connections between linear and linear layers

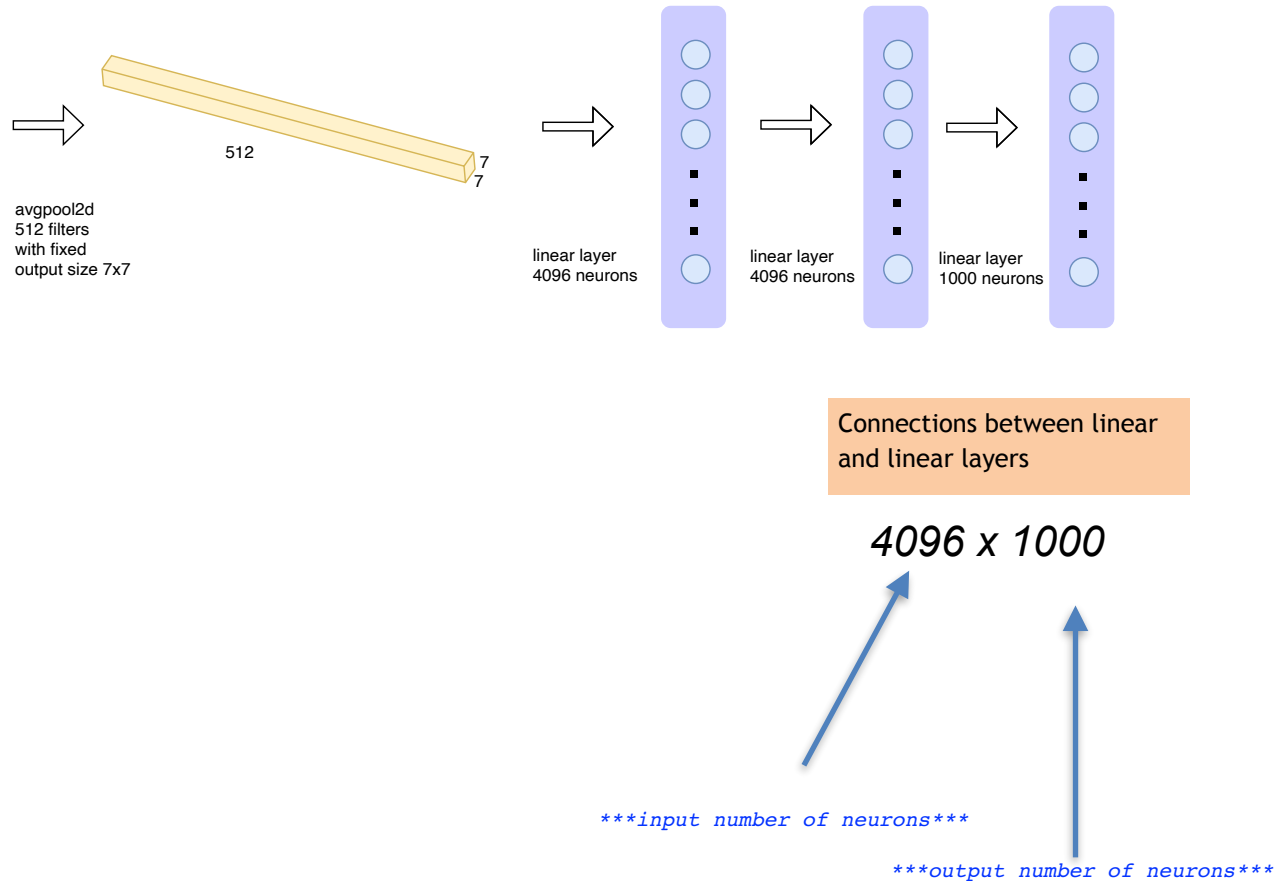
4096×4096

input number of neurons

output number of neurons

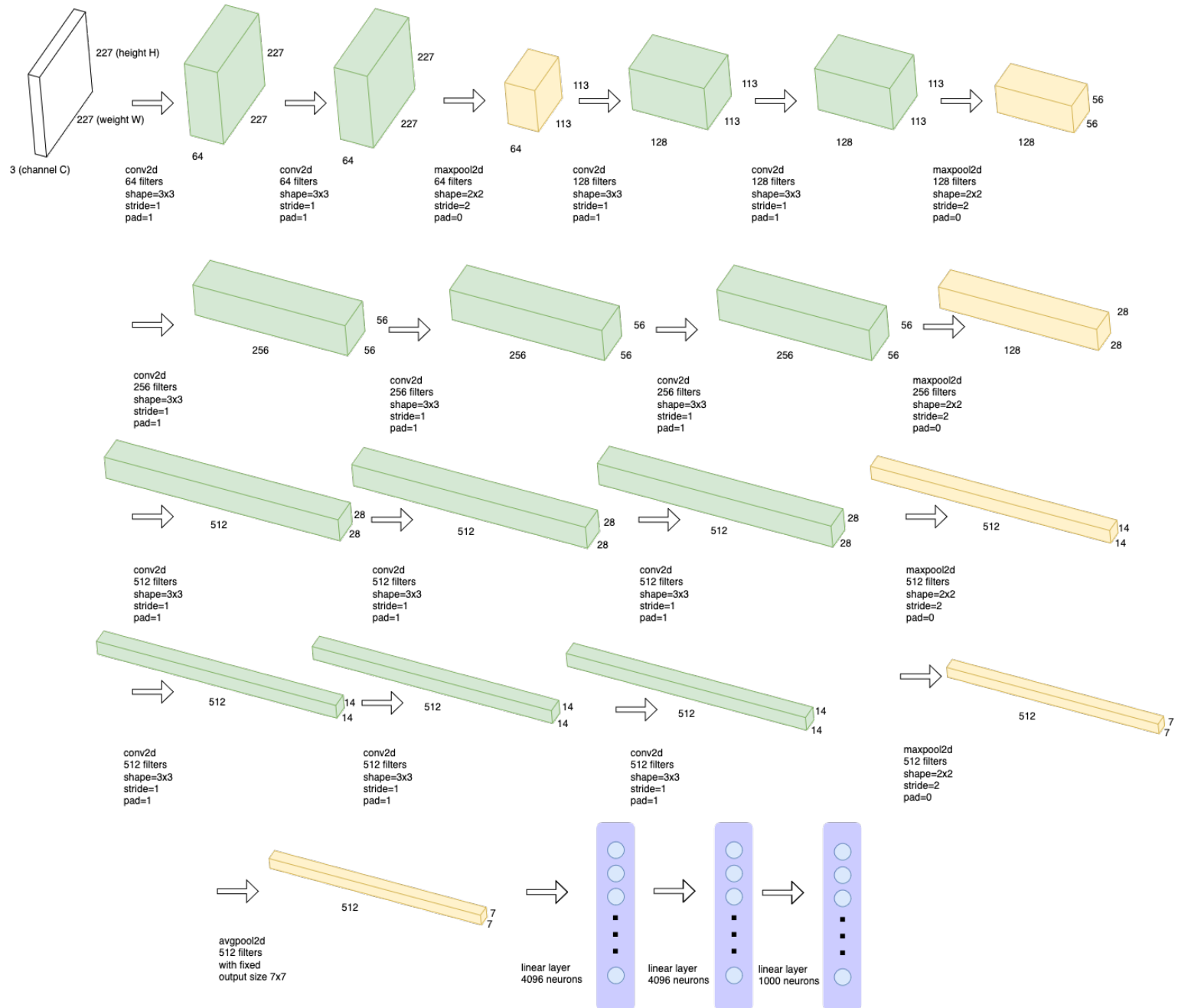
Solution: Group Activity

What are the dimensions of different volumes?



Full solution: Group Activity

What are the dimensions of different volumes?



Popular CNN: VGG-16

```
[torch.Size([3, 227, 227])] INPUT
[torch.Size([64, 227, 227])] CONV1_1: 64 convolutional filters each with a shape=3x3, stride=1, pad=1
[torch.Size([64, 227, 227])] CONV1_2: 64 convolutional filters each with a shape=3x3, stride=1, pad=1
[torch.Size([64, 113, 113])] MAX POOL1: 2x2 max pooling filters with stride=2, pad=0
[torch.Size([128, 113, 113])] CONV2_1: 128 convolutional filters each with a shape=3x3, stride=1, pad=1
[torch.Size([128, 113, 113])] CONV2_2: 128 convolutional filters each with a shape=3x3, stride=1, pad=1
[torch.Size([128, 56, 56])] MAX POOL2: 2x2 max pooling filters with stride=2, pad=0
[torch.Size([256, 56, 56])] CONV3_1: 256 convolutional filters each with a shape=3x3, stride=1, pad=1
[torch.Size([256, 56, 56])] CONV3_2: 256 convolutional filters each with a shape=3x3, stride=1, pad=1
[torch.Size([256, 56, 56])] CONV3_3: 256 convolutional filters each with a shape=3x3, stride=1, pad=1
[torch.Size([256, 28, 28])] MAX POOL3: 2x2 max pooling filters with stride=2, pad=0
[torch.Size([512, 28, 28])] CONV4_1: 512 convolutional filters each with a shape=3x3, stride=1, pad=1
[torch.Size([512, 28, 28])] CONV4_2: 512 convolutional filters each with a shape=3x3, stride=1, pad=1
[torch.Size([512, 28, 28])] CONV4_3: 512 convolutional filters each with a shape=3x3, stride=1, pad=1
[torch.Size([512, 14, 14])] MAX POOL4: 2x2 max pooling filters with stride=2, pad=0
[torch.Size([512, 14, 14])] CONV5_1: 512 convolutional filters each with a shape=3x3, stride=1, pad=1
[torch.Size([512, 14, 14])] CONV5_2: 512 convolutional filters each with a shape=3x3, stride=1, pad=1
[torch.Size([512, 14, 14])] CONV5_3: 512 convolutional filters each with a shape=3x3, stride=1, pad=1
[torch.Size([512, 7, 7])] MAX POOL5: 2x2 max pooling filters with stride=2, pad=0
[torch.Size([512, 7, 7])] AVG POOL: fixed output size [width x height]
[torch.Size([4096])] FC6: neurons
[torch.Size([4096])] FC7: neurons
[torch.Size([1000])] FC8: neurons
```

ImageNet winners by the popular CNNs

- AlexNet (2012) → VGG (2014) → ResNet (2015)

