# CS 6824:
# Convolutional Neural Networks:
# The Inflection Point

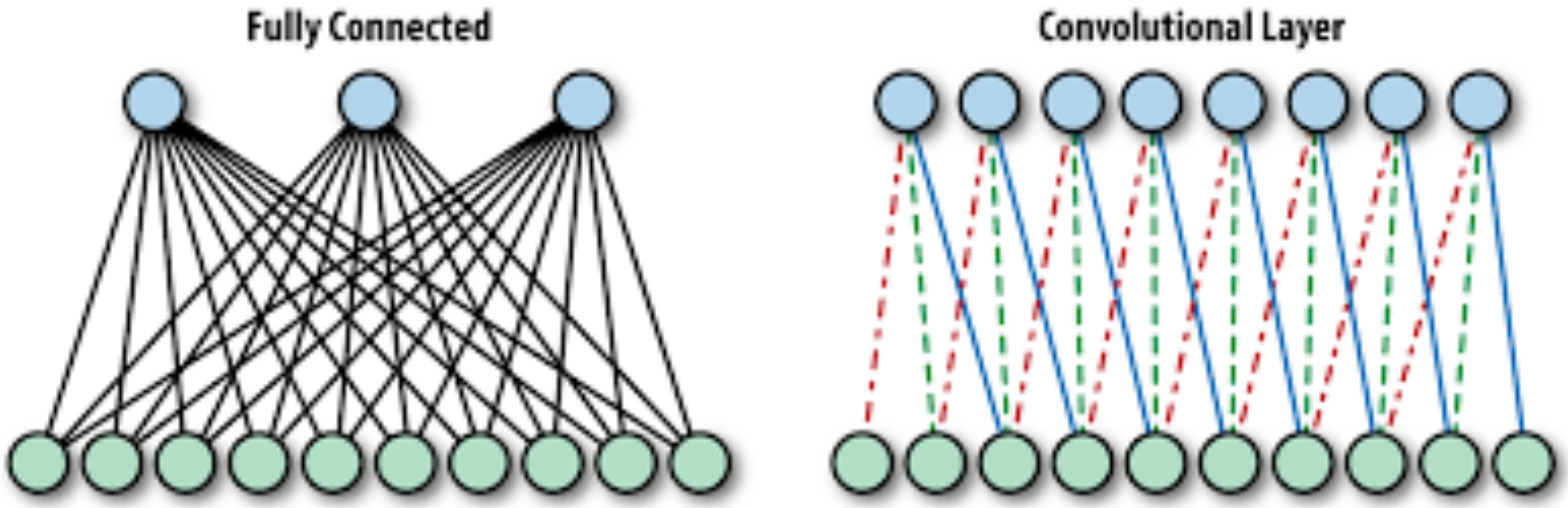
**Acknowledgement**:
Many of these slides are derived from Tom Mitchell, Pascal Poupart, Pieter Abbeel, Eric Eaton, Carlos Guestrin, William Cohen, and Andrew Moore.

# CNNs

- Convolutional neural networks have gained a special status over the last few years as an especially promising form of deep learning. Rooted in image processing, convolutional layers have found their way into virtually all subfields of deep learning, and are very successful for the most part.

- While small and fast, the CNNs are highly representative of the type of models used in practice to obtain state-of-the-art results in object-recognition tasks.

# Fully connected NN vs CNN

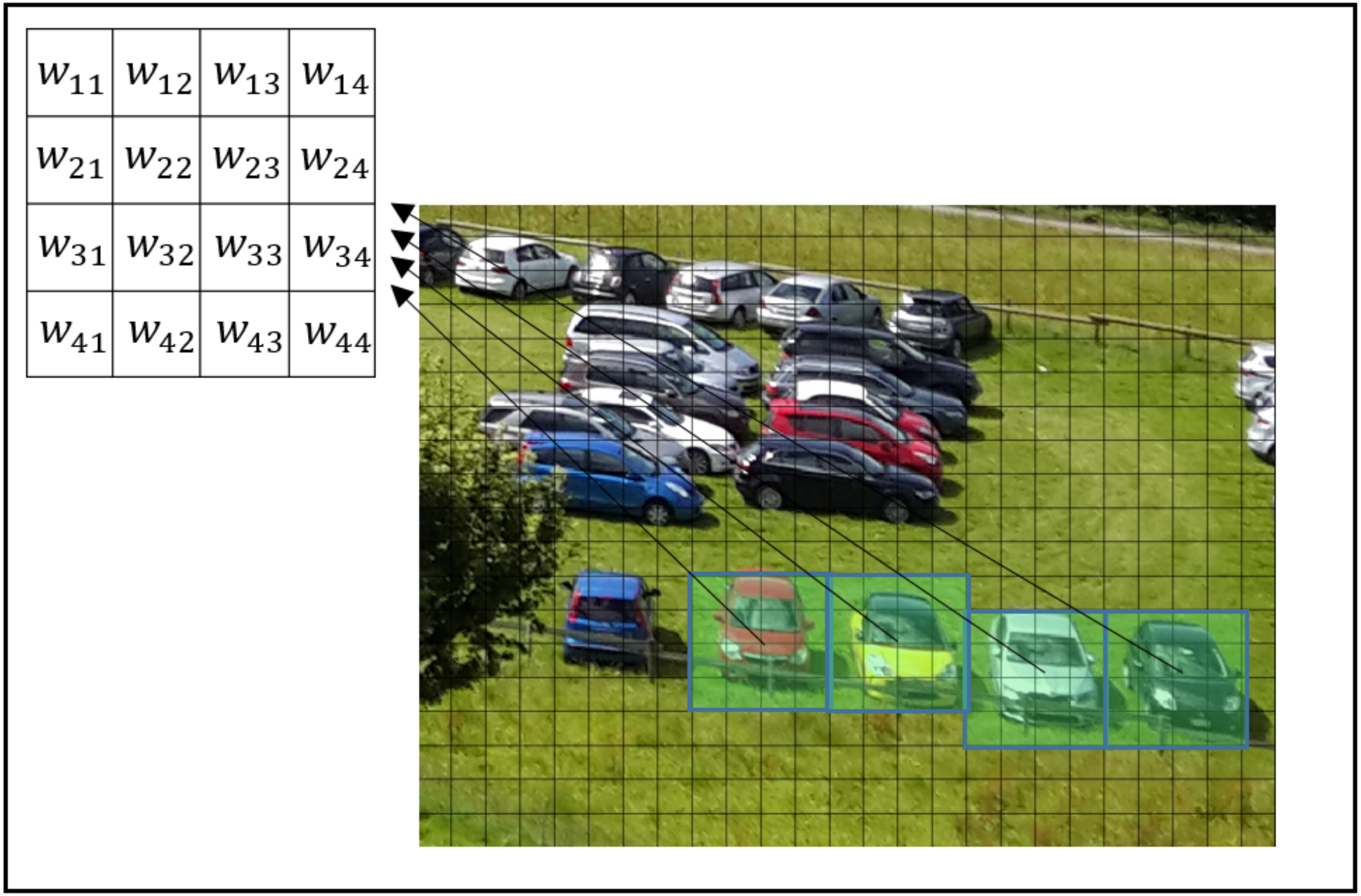


The fundamental difference between *fully connected* and *convolutional* neural networks is the pattern of connections between consecutive layers. In the fully connected case, as the name might suggest, each unit is connected to all of the units in the previous layer.

In a convolutional layer of a neural network, on the other hand, each unit is connected to a (typically small) number of nearby units in the previous layer. Furthermore, all units are connected to the previous layer in the same way, with the exact same weights and structure. This leads to an operation known as *convolution*, giving the architecture its name.

# Advantage of CNN - shared weights

# CNN for an image



| | | | |
|---|---|---|---|
| $w_{11}$ | $w_{12}$ | $w_{13}$ | $w_{14}$ |
| $w_{21}$ | $w_{22}$ | $w_{23}$ | $w_{24}$ |
| $w_{31}$ | $w_{32}$ | $w_{33}$ | $w_{34}$ |
| $w_{41}$ | $w_{42}$ | $w_{43}$ | $w_{44}$ |

The convolutional filter—a "sliding window"—applied across an image.

# Convolutions for feature extraction

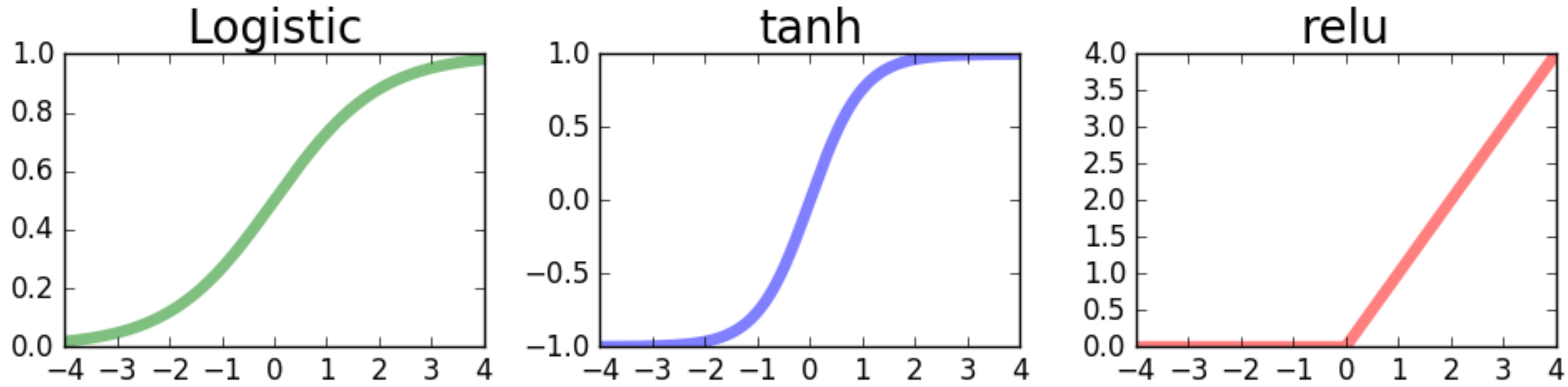
- In neural networks
  - A convolution denotes the linear combination of a subset of units based on a specific pattern of weights.

$$a_j = \sum_i w_{ji} z_i$$

  - Convolutions are often combined with an activation function to produce a feature

$$z_j = h(a_j) = h\left(\sum_i w_{ji} z_i\right)$$

# Activation Functions



$$h(a) = \sigma(a) = \frac{1}{1 + e^{-a}}$$

$$h(a) = tanh(a) = \frac{e^{a} - e^{-a}}{e^{a} + e^{-a}}$$

$$h(a) = \max(0,a)$$

# Convolution Neural Network (CNN)

- A **CNN** refers to any network that consists of an **alternation of convolution and pooling layers**, where **some of the convolution weights are shared**
- Architecture:

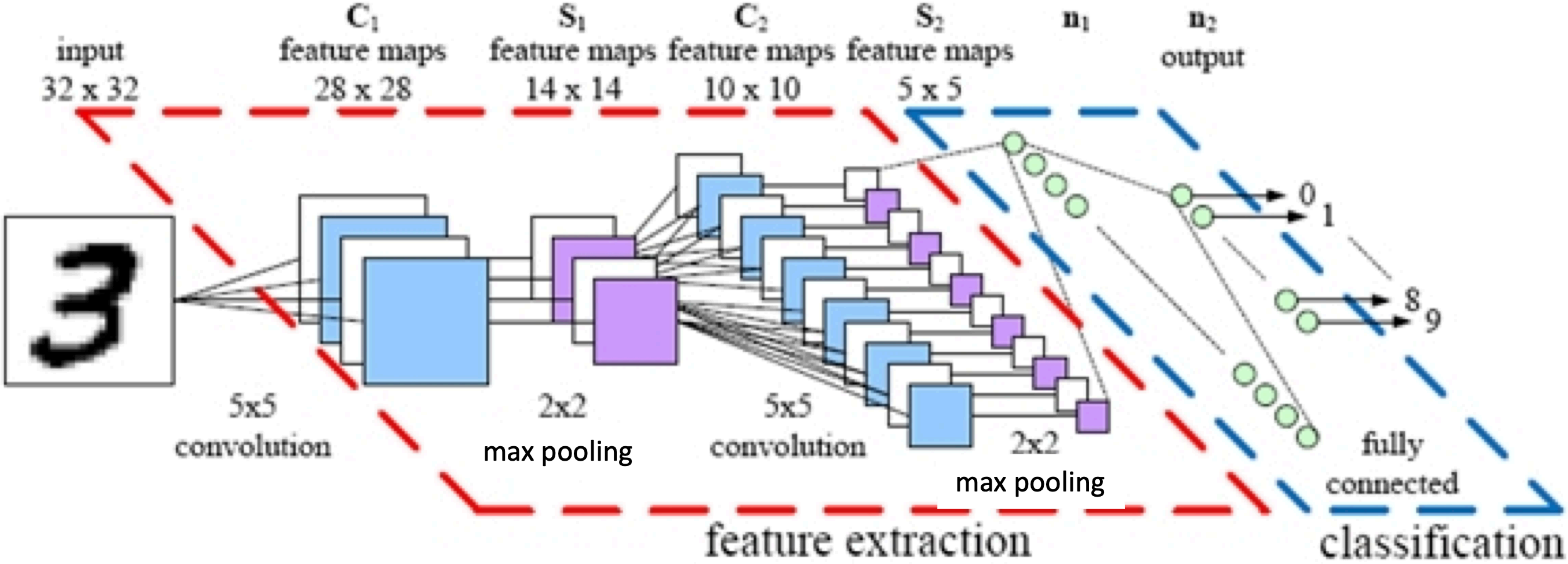
# Pooling

- Pooling: **commutative** mathematical operation that combines several units

- Examples:
  - max, sum, product, average, Euclidean norm, etc.

- Commutative property (order does not matter):
  - max($a$, $b$) = max($b$, $a$)

# Digit Recognition



# Benefits of CNN

- Sparse interactions
  - Fewer connections

- Parameter sharing
  - Fewer weights

- Locally equivariant representation
  - Locally invariant to translations
  - Handle inputs of varying length

# Parameters

- # **of filters**: integer indicating the #of filters applied to each window

- **kernel size**: tuple (width, height) indicating the size of the window

- **Stride**: tuple (horizontal, vertical) indicating the horizontal and vertical shift between each window

- **Padding**: “valid” or ”same”. Valid indicates no input padding. Same indicates that the input is padded with a border of zeros to ensure that the output has the same size as the input

# Examples

# Training CNN

- Convolutional neural networks are trained in the same way as other neural networks through backpropagation
  - AdaGrad, RMSprop, Adam

- Weight sharing:
  - Combine gradients of shared weights into a single gradient

# Architecture design

- What is the preferred filter size?
- ]VGG (Visual Geometry Group at Oxford, 2014): stack of small filters is often preferred to single large filter
  - Fewer parameters
  - Deeper network
- Schematic:

# Residual Networks

- **Idea**: Addressing vanishing gradient problem by introducing residual connections (a.k.a. skip connections) to shorten paths (He et al. 2015)

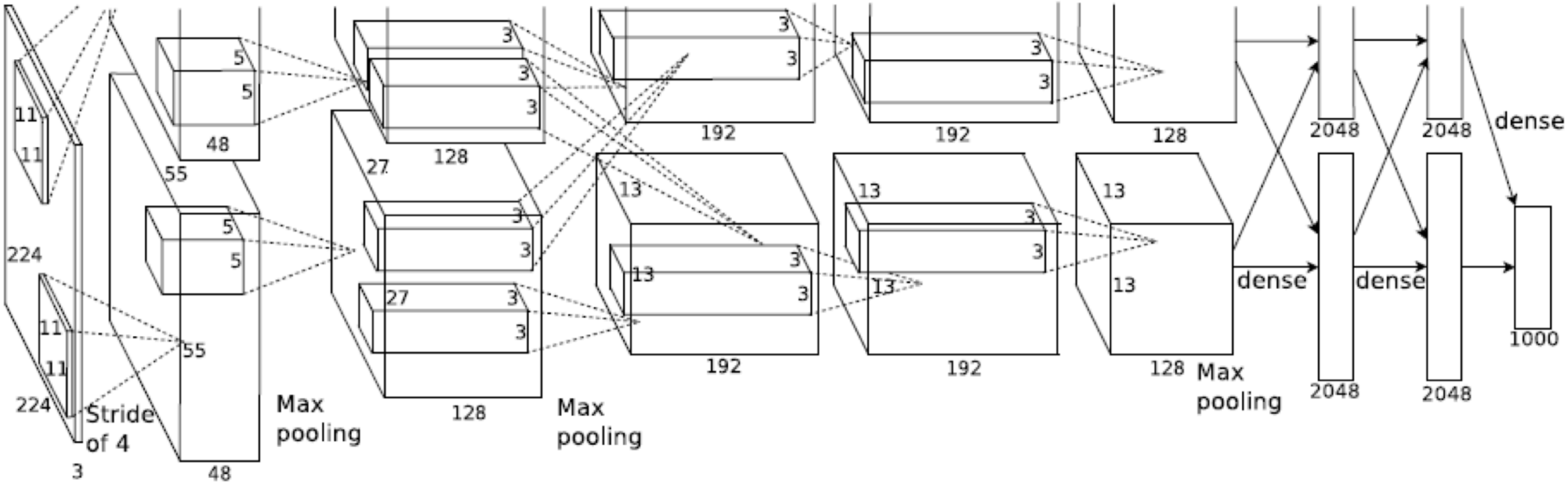
- Schematic:

# Applications

- Speech Recognition
- **Image recognition**
- Machine translation
- Control
- …
- Data with sequential, spatial or tensor patterns

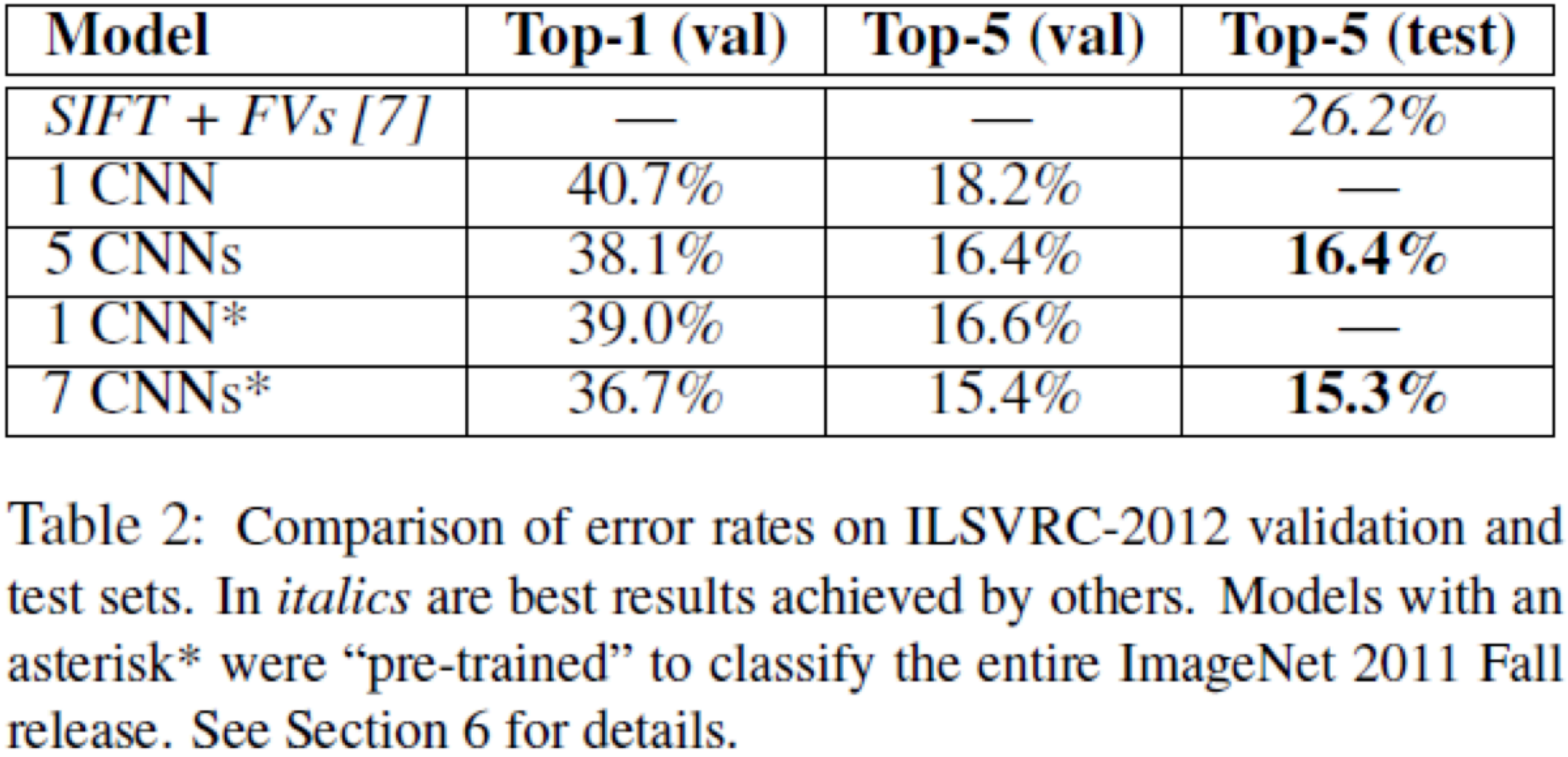
# Image Recognition

- Convolutional Neural Network
  - With rectified linear units and dropout
  - Data augmentation for transformation invariance



# ImageNet Breakthrough

- Results: ILSVRC-2012
  - Krizhevsky, Sutskever, Hinton

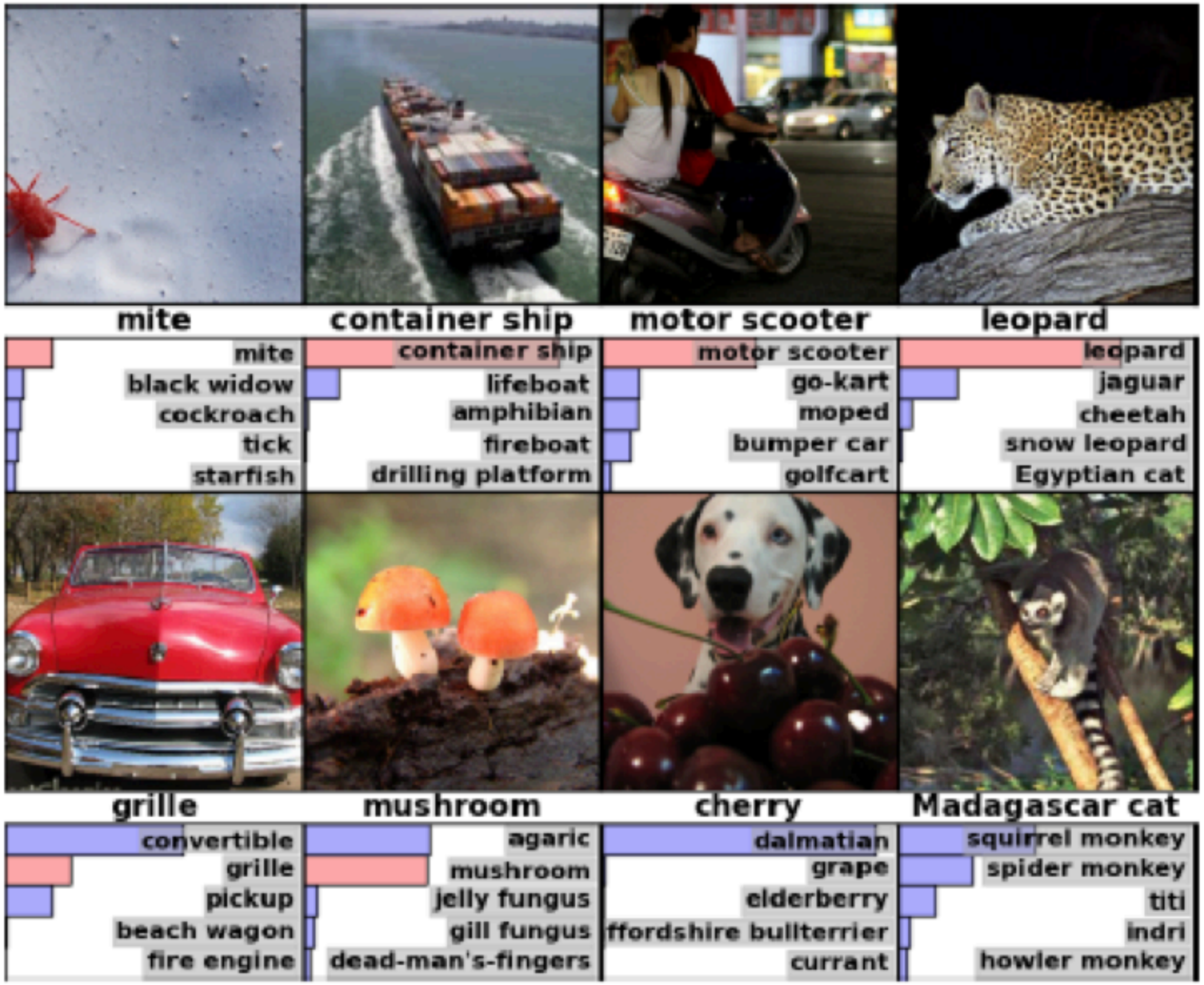


| Model | Top-1 (val) | Top-5 (val) | Top-5 (test) |
|---|---|---|---|
| *SIFT + FVs [7]* | — | — | *26.2%* |
| 1 CNN | 40.7% | 18.2% | — |
| 5 CNNs | 38.1% | 16.4% | **16.4%** |
| 1 CNN* | 39.0% | 16.6% | — |
| 7 CNNs* | 36.7% | 15.4% | **15.3%** |

Table 2: Comparison of error rates on ILSVRC-2012 validation and test sets. In *italics* are best results achieved by others. Models with an asterisk* were "pre-trained" to classify the entire ImageNet 2011 Fall release. See Section 6 for details.

# ImageNet Breakthrough

- From Krizhevsky, Sutskever, Hinton


mite
container ship
motor scooter
leopard
mite
black widow
cockroach
tick
starfish
container ship
lifeboat
amphibian
fireboat
drilling platform
motor scooter
go-kart
moped
bumper car
golfcart
leopard
jaguar
cheetah
snow leopard
Egyptian cat
grille
mushroom
cherry
Madagascar cat
convertible
grille
pickup
beach wagon
fire engine
agaric
mushroom
jelly fungus
gill fungus
dead-man's-fingers
dalmatian
grape
elderberry
ffordshire bullterrier
currant
squirrel monkey
spider monkey
titi
indri
howler monkey