# CNN Architectures

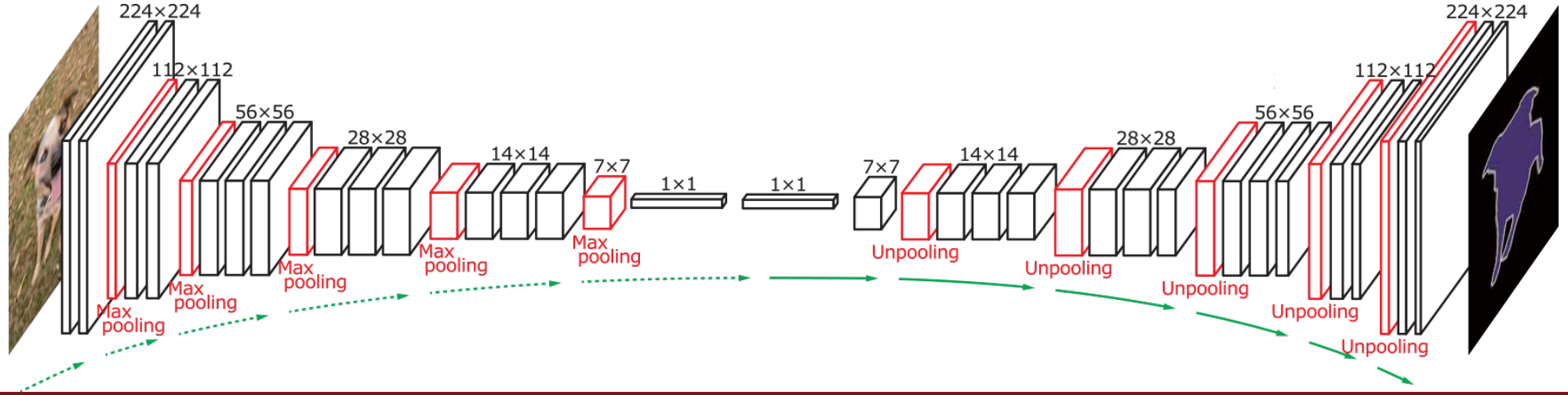
COSC 410: Applied Machine Learning

Spring 2022

Prof. Apthorpe

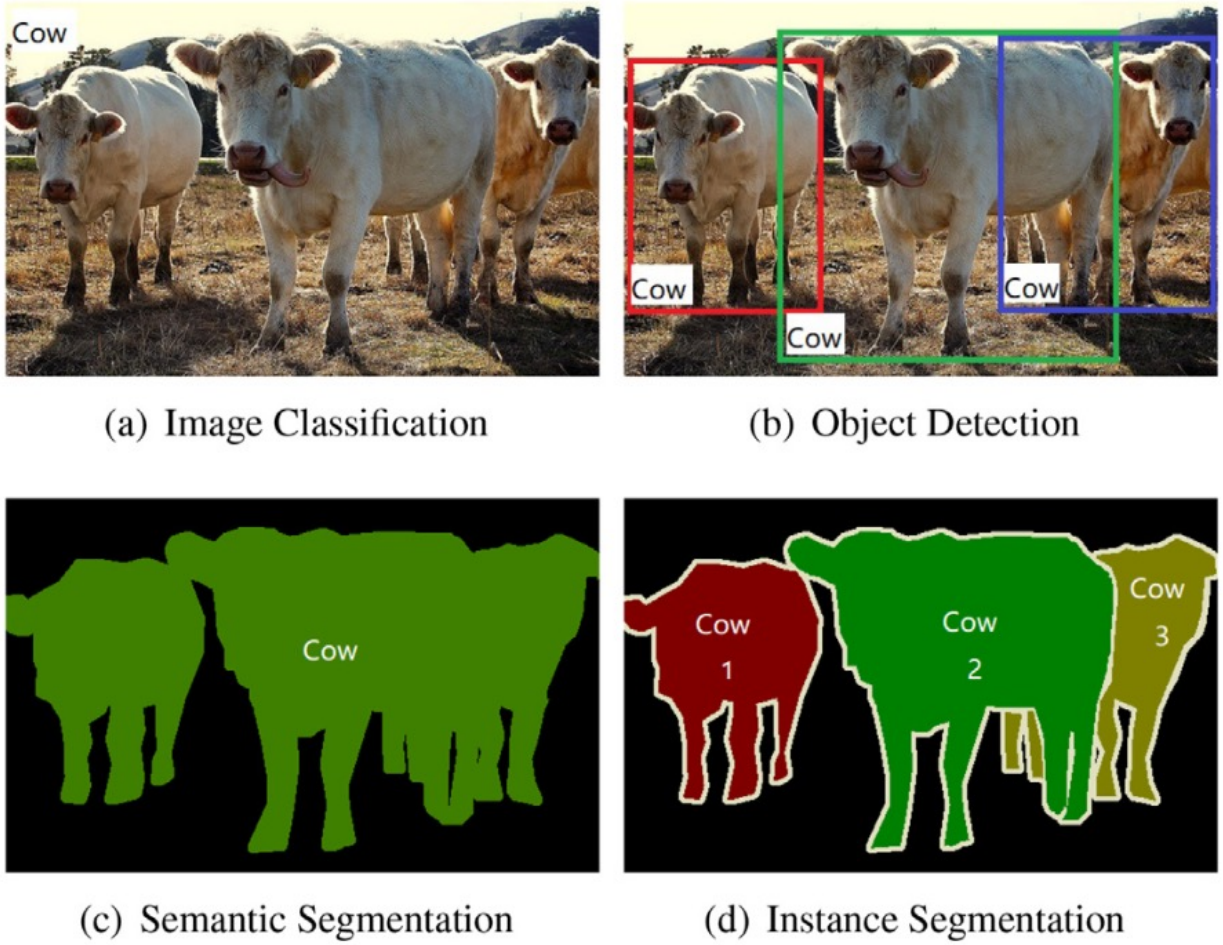
# Outline

- LeNet-style CNN Architectures
- Counting CNN Parameters
- Other Popular Architectures
- Deconvolutions & U-Nets
- 1D & 3D CNNs



# Computer Vision Tasks

- **Image classification**
  - Predict labels per **image**
- **Object Detection**
  - Predict labels & bounding boxes
- **Image segmentation**
  - Predict labels per **pixel**



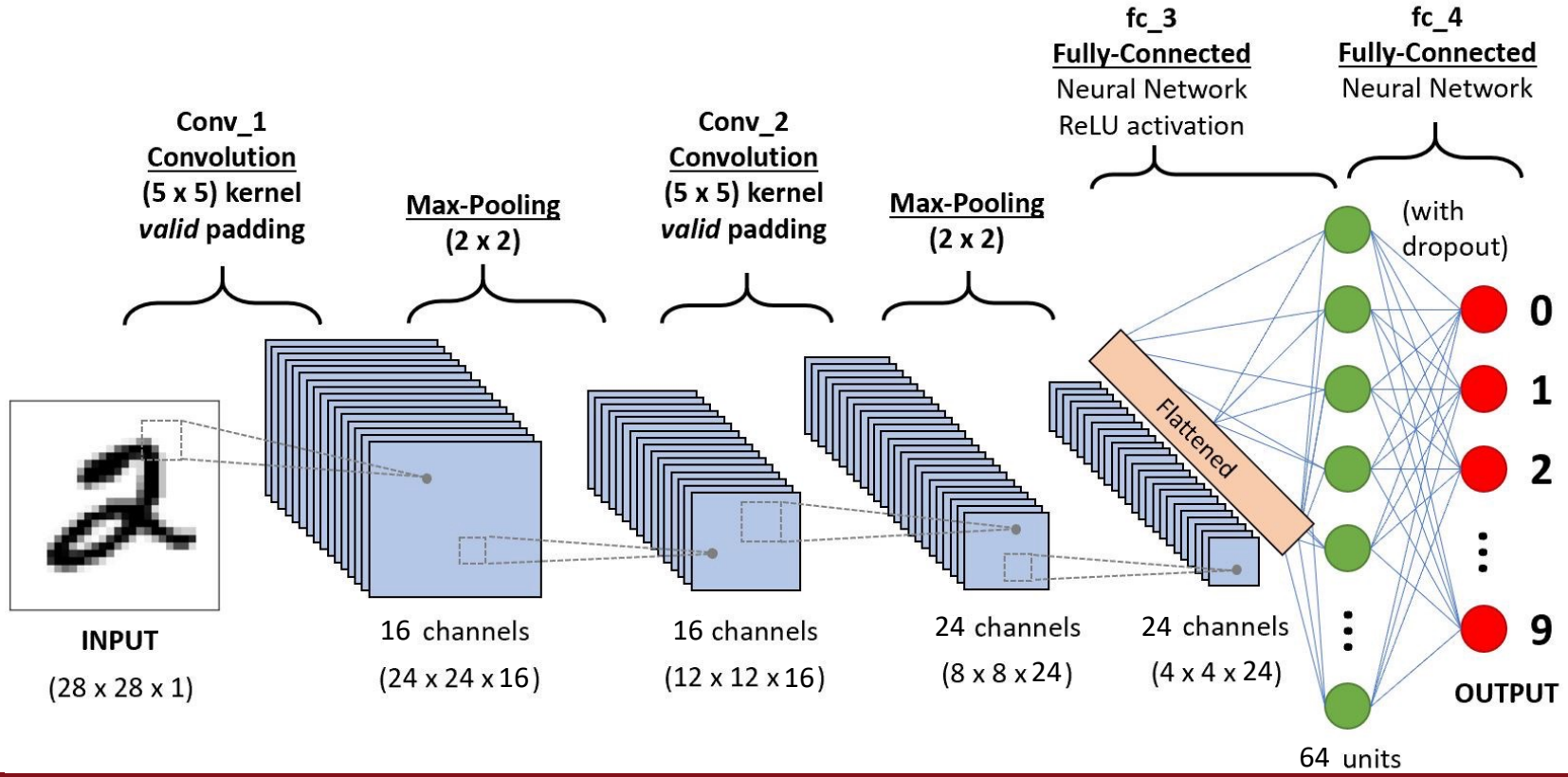

(a) Image Classification (b) Object Detection

(c) Semantic Segmentation (d) Instance Segmentation

# LeNet-style CNN Architecture

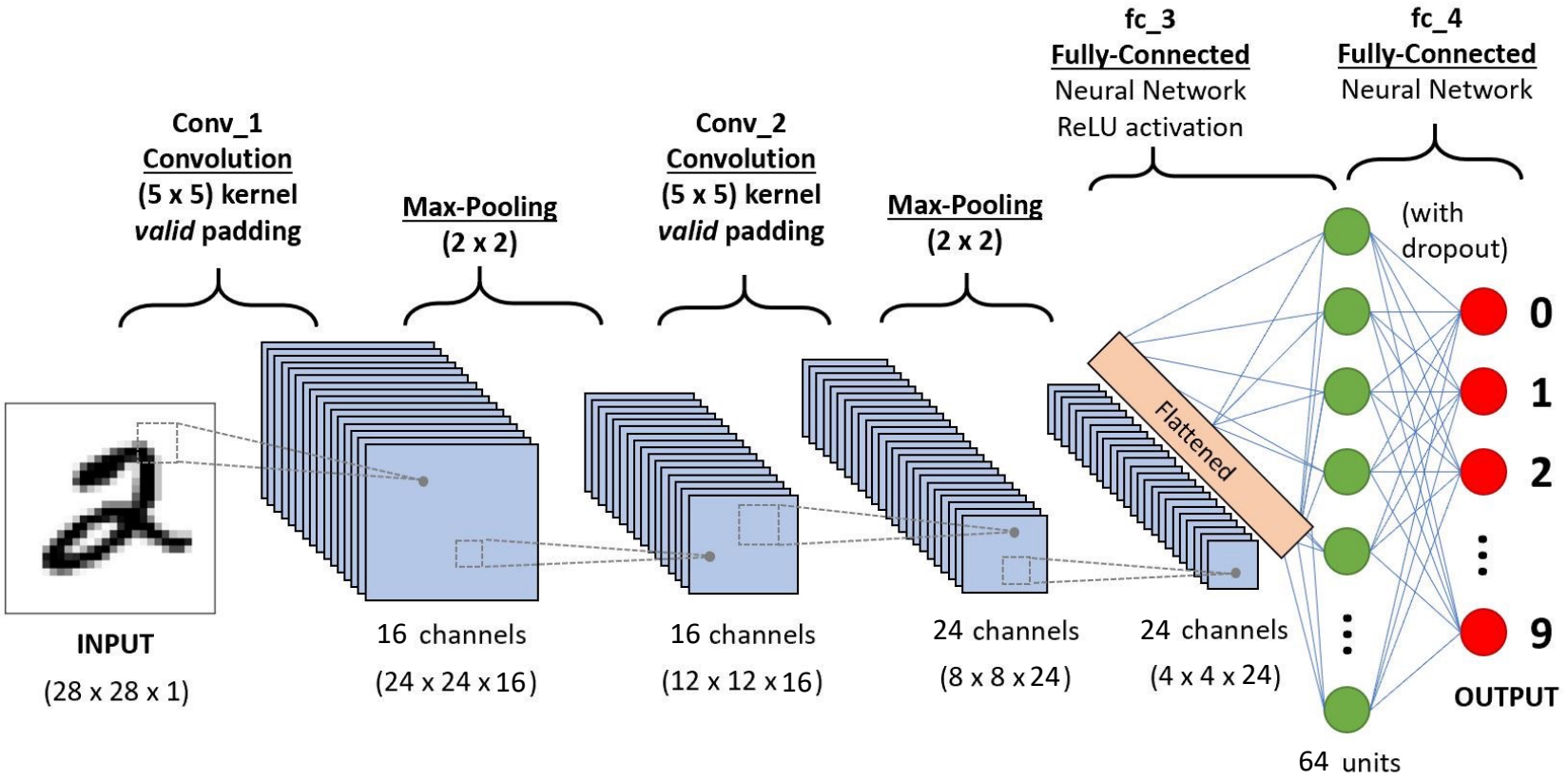
Original LeNet proposed in 1989 by Yann LeCun, et al.

- Convolution & pooling with increasing # of filters → FNN layers



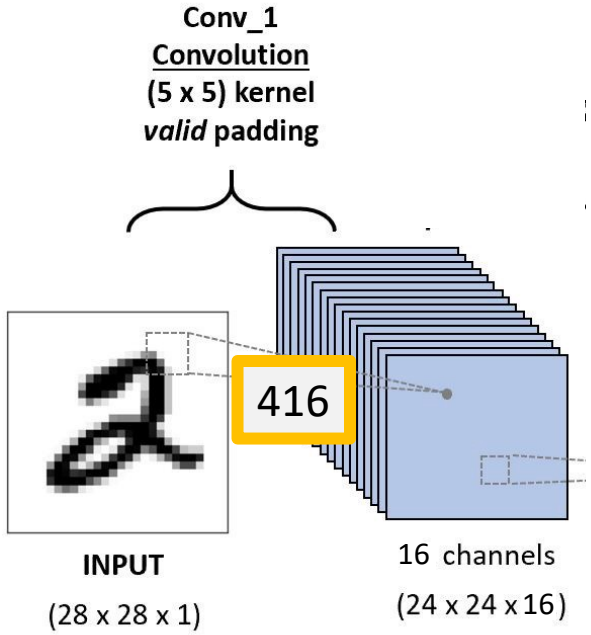
# CNN Parameter Counts

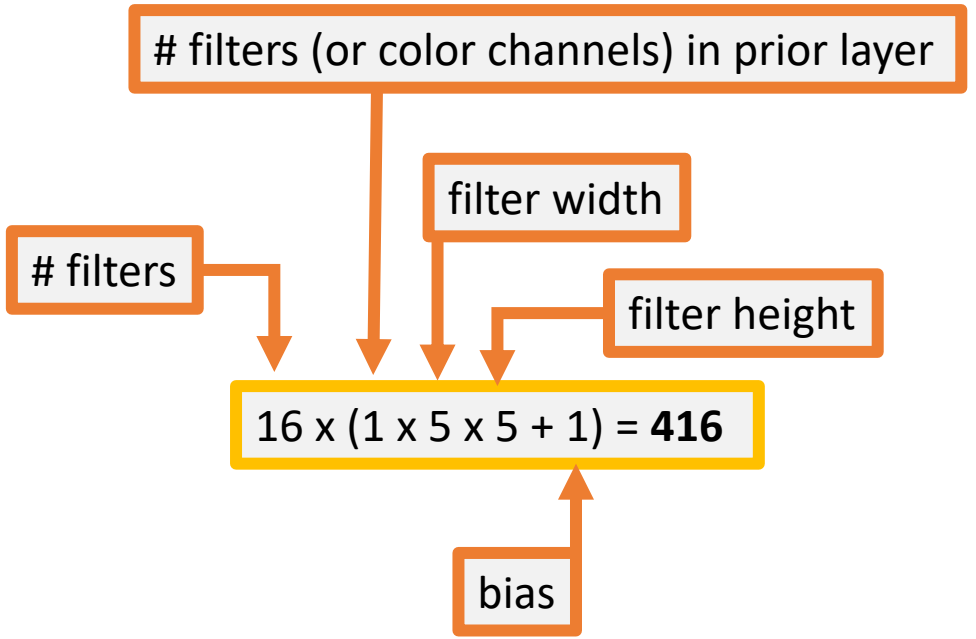
- How many parameters are in *each layer* of this network?



# CNN Parameter Counts

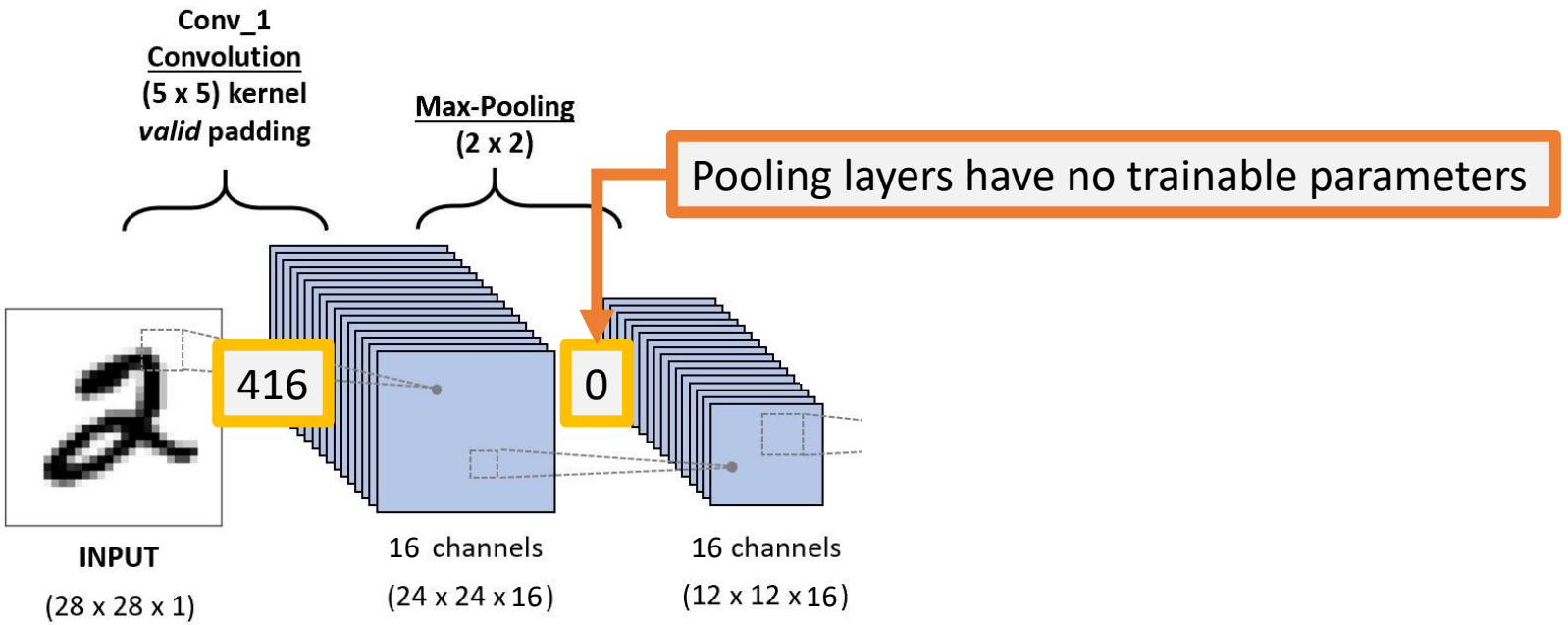
- How many parameters are in *each layer* of this network?



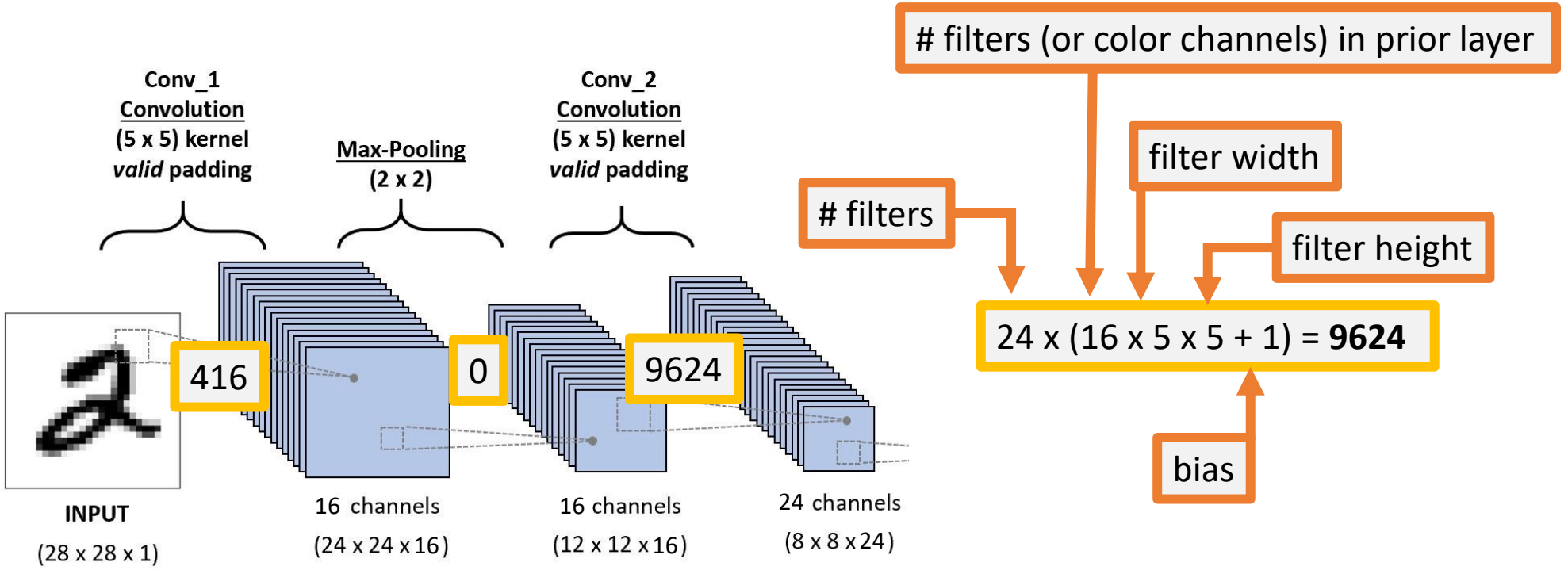
# CNN Parameter Counts

- How many parameters are in *each layer* of this network?



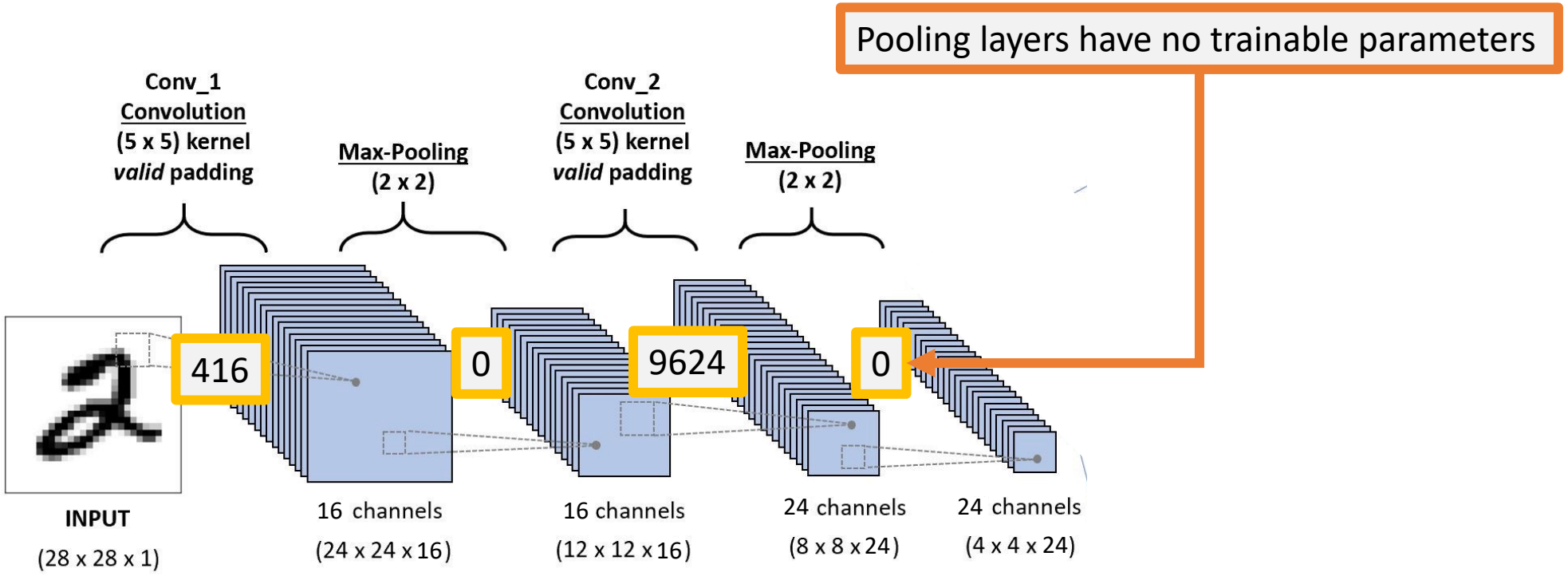
# CNN Parameter Counts

- How many parameters are in *each layer* of this network?



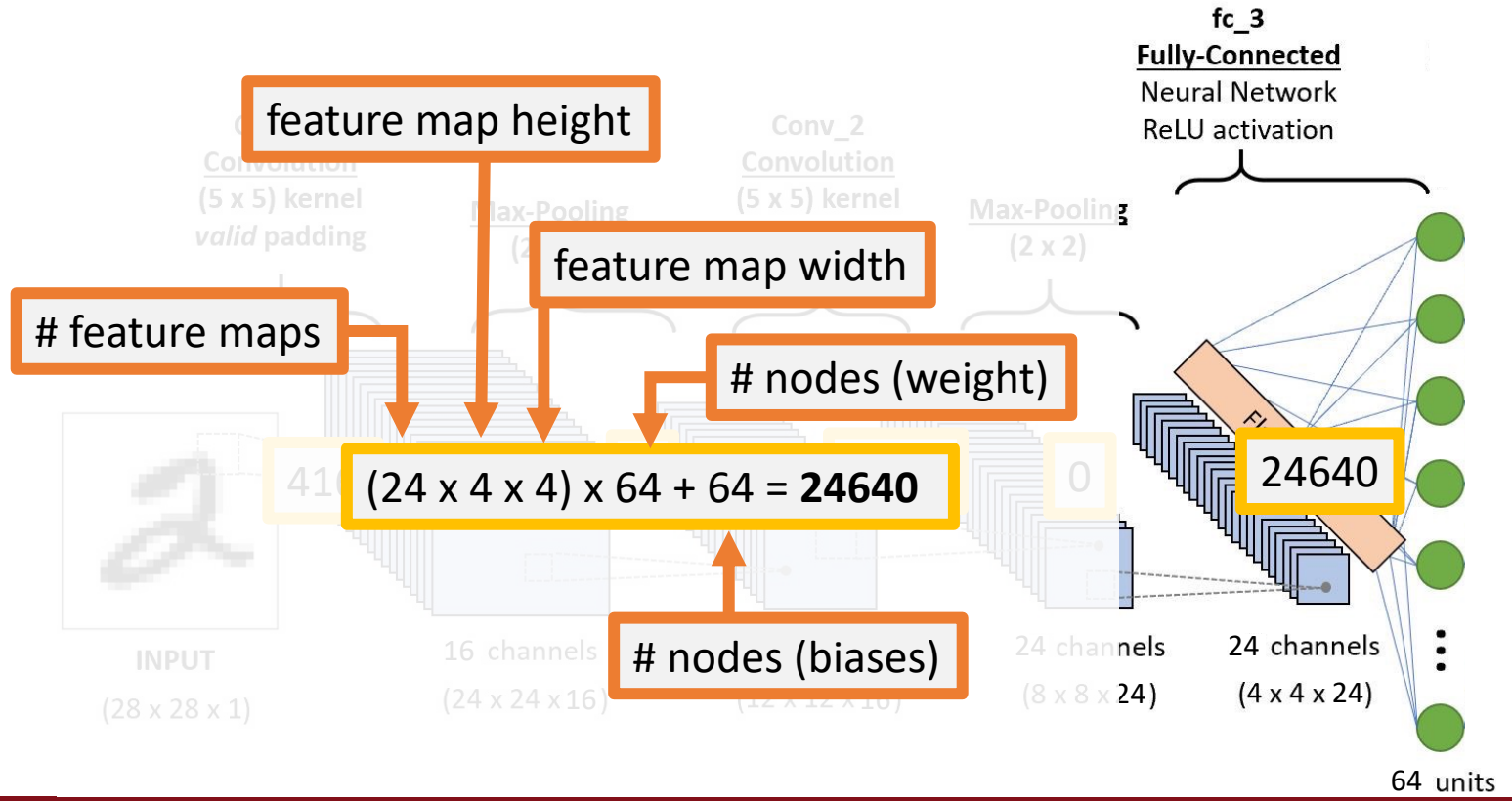
# CNN Parameter Counts

- How many parameters are in *each layer* of this network?



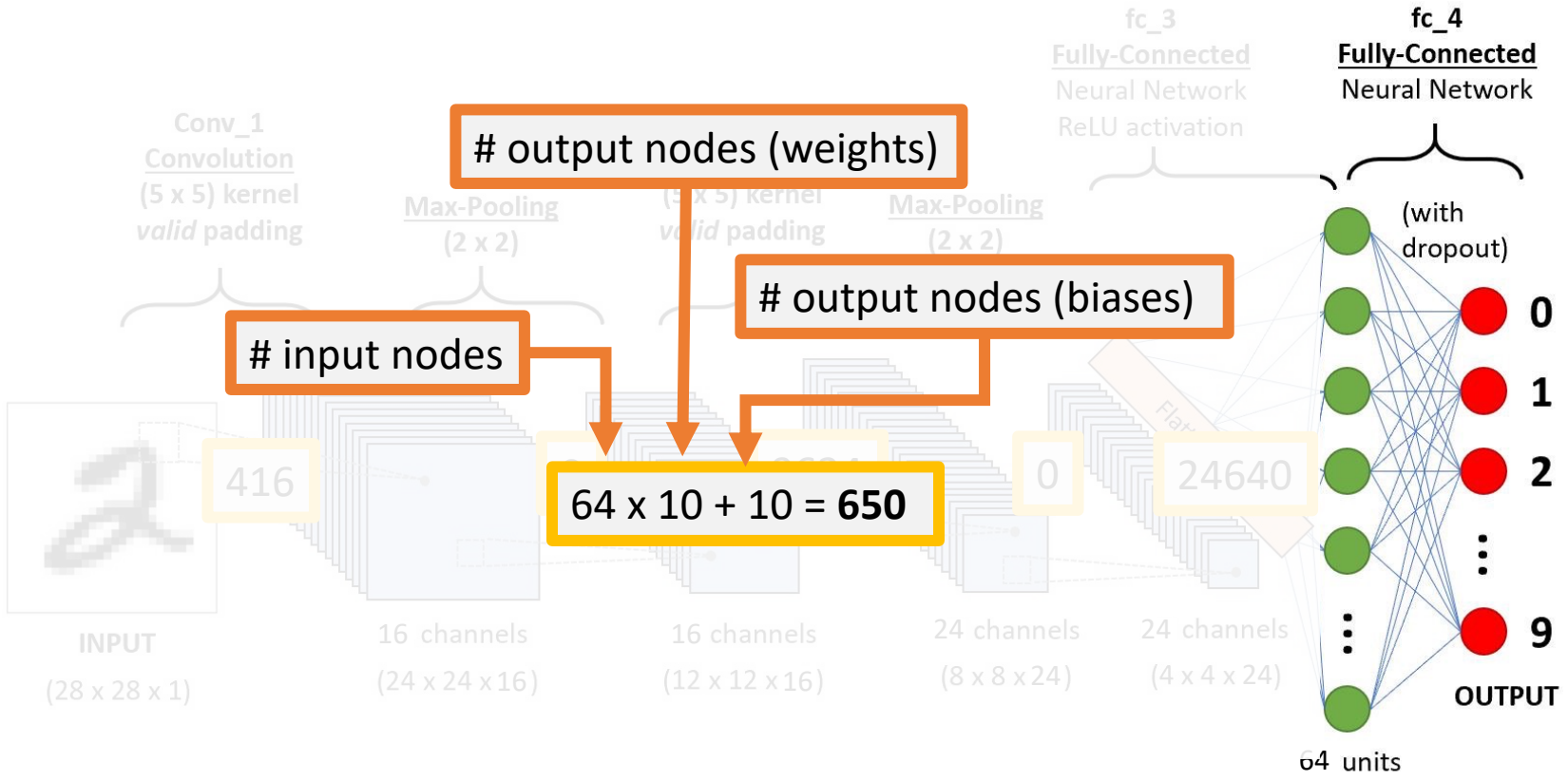
# CNN Parameter Counts

- How many parameters are in *each layer* of this network?



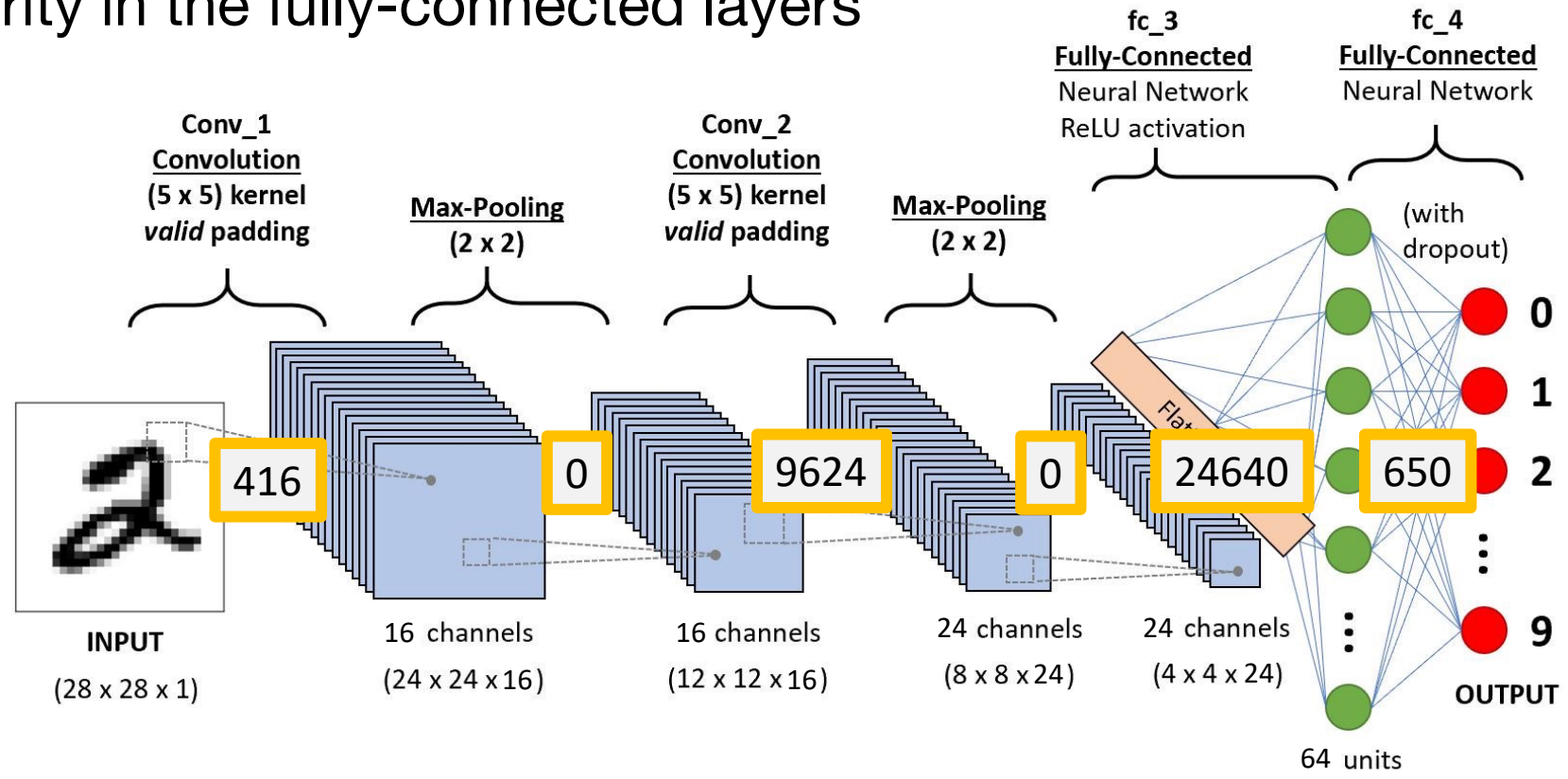
# CNN Parameter Counts

- How many parameters are in *each layer* of this network?



# CNN Parameter Counts

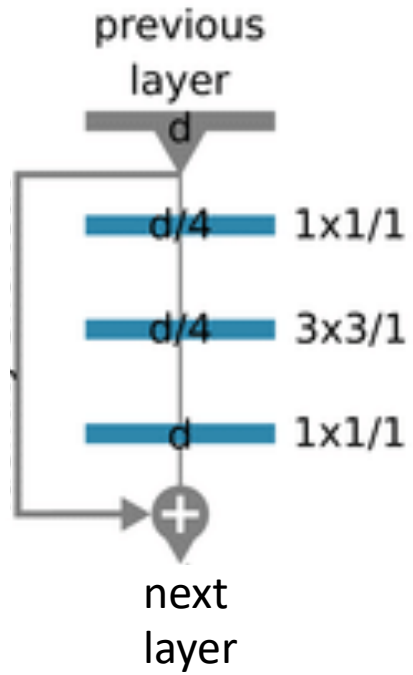
- **35,330** total parameters
  - ★ Majority in the fully-connected layers



# Popular CNN Architectures

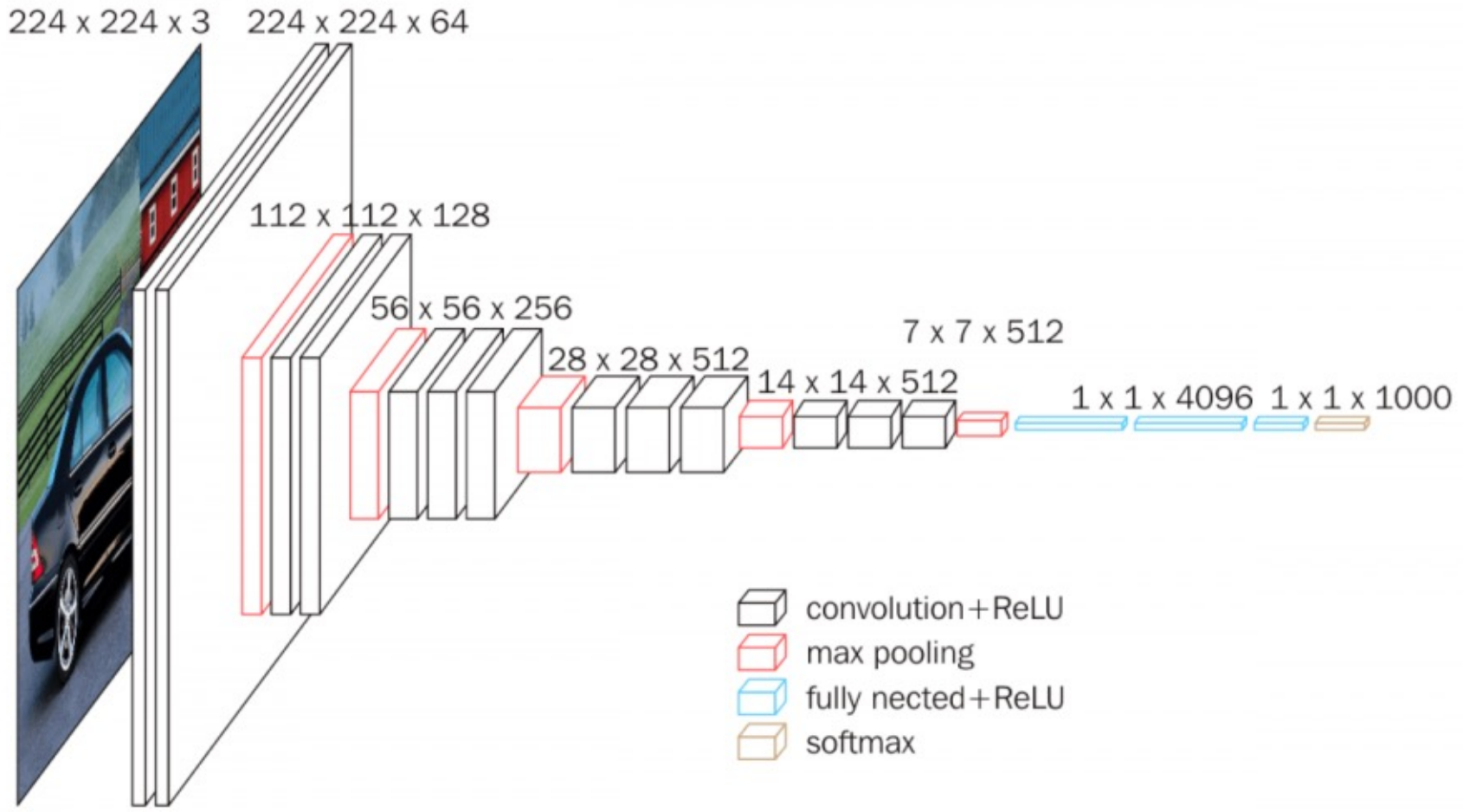
- Many CNN architectures have been designed for different tasks
- Trends:
  - Smaller filters (e.g. 3x3)
  - More layers
  - Residual & skip connections
    - Same philosophy as “wide & deep” networks

**Residual Connection**



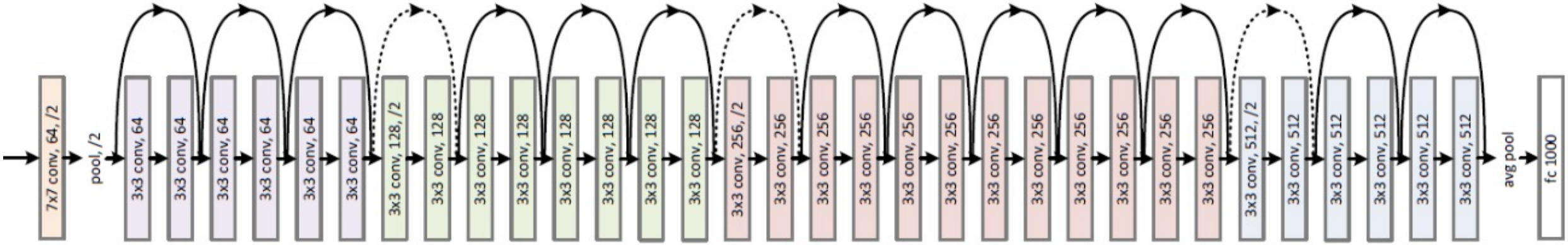
# VGG-16

VGG stands for **"Visual Geometry Group"** after the group of researchers at Oxford who designed the network

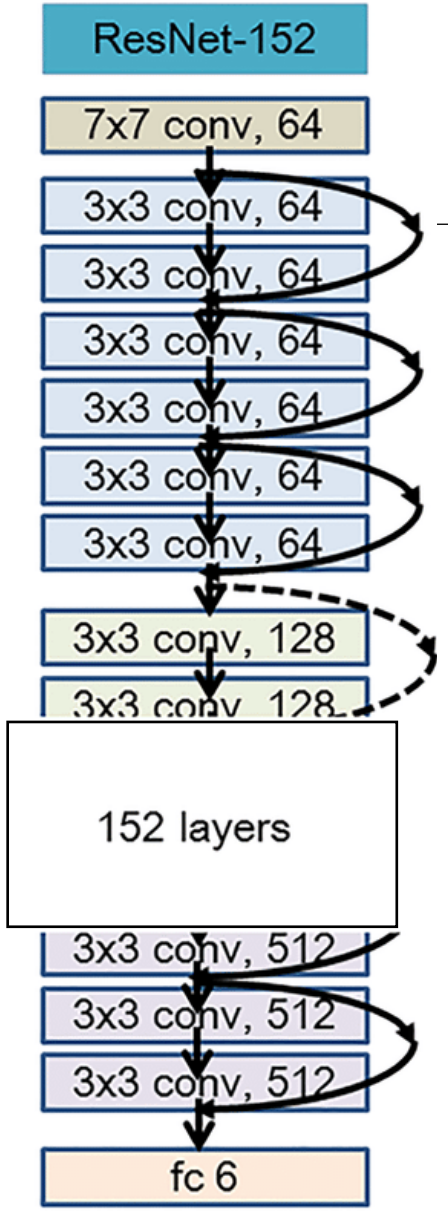


# ResNet-34

ResNet is named after the many **residual connections** in the network

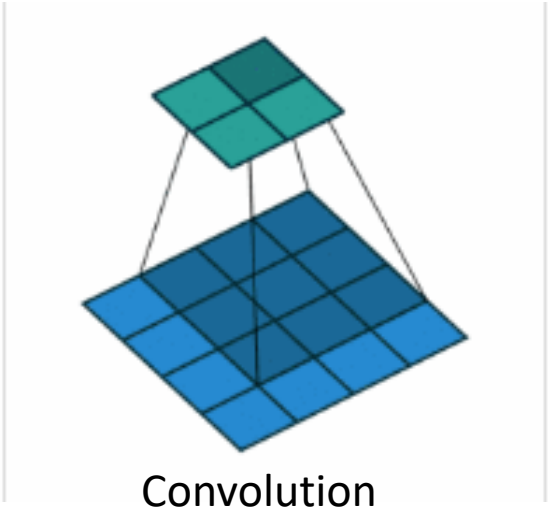


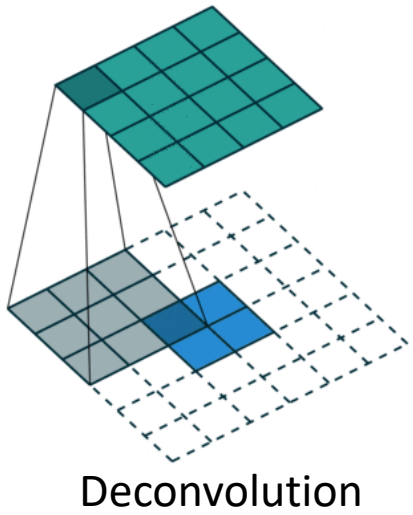
# ResNet-152


ResNet-152
7x7 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 128
3x3 conv, 128
152 layers
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
fc 6

# Deconvolutions

- Many CNNs for image **segmentation** include **deconvolutional layers** to produce output that is the same X & Y size as the input with per-pixel labels
  - Deconvolutional layers are **up-sampling** steps that reverse the down-sampling size reductions performed by pooling layers



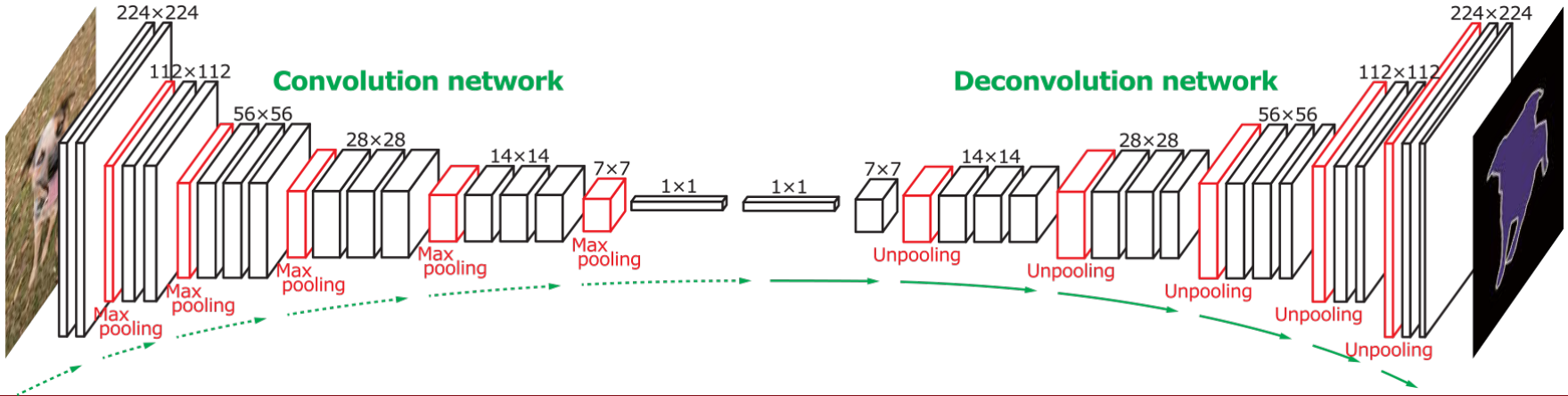




Keras `Conv2DTranspose` layer implements a 2D deconvolution

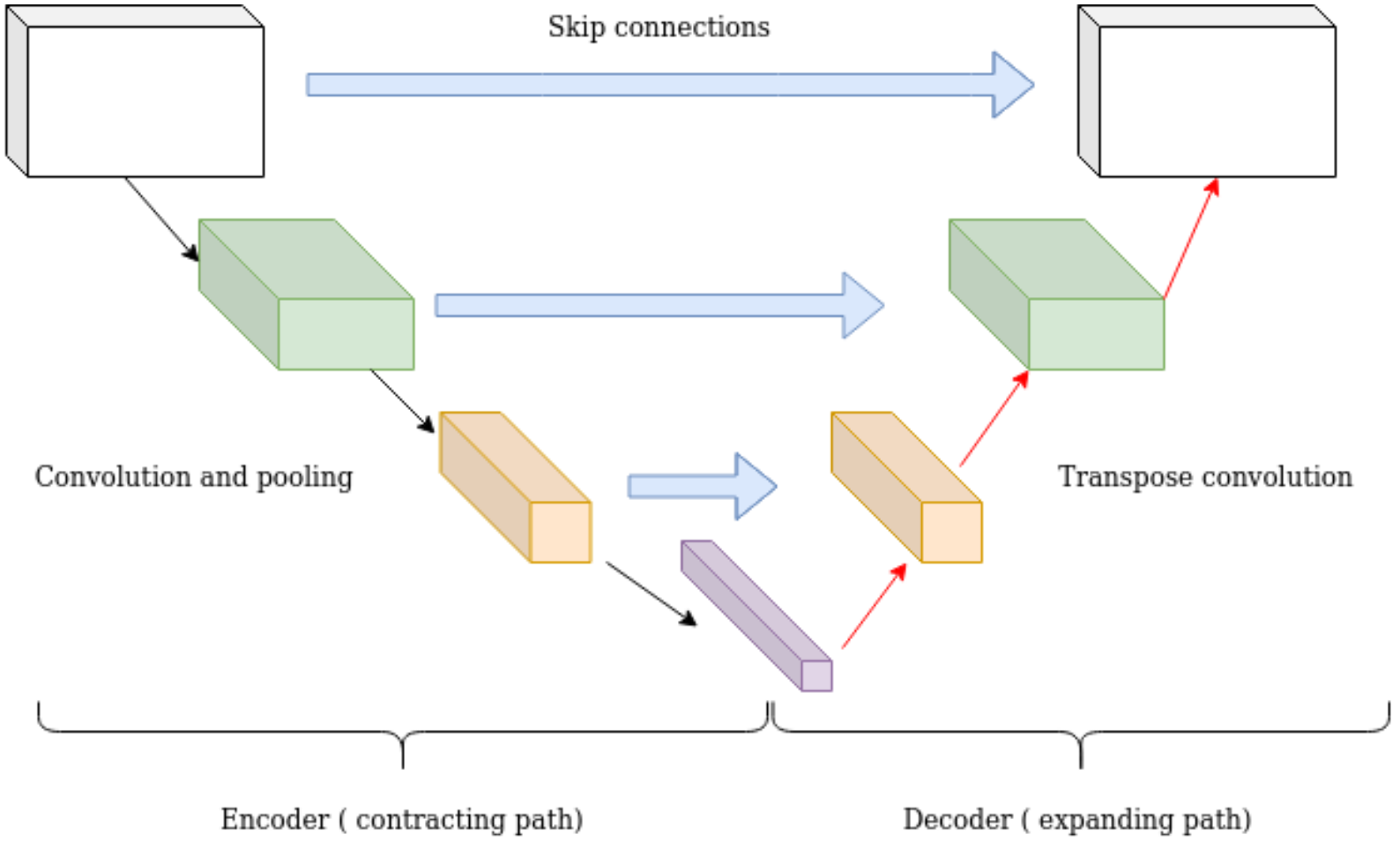
# U-Nets

**Deconvolution**: Upscaling with trainable filter weights
**Unpooling**: simple upscaling, just repeat the pixel value NxN times

- Common architecture for image segmentation
  - Series of convolution & pooling layers → more, smaller feature maps
  - Series of convolution & deconvolution (or unpooling) layers → fewer larger feature maps
- Output is same size as original image

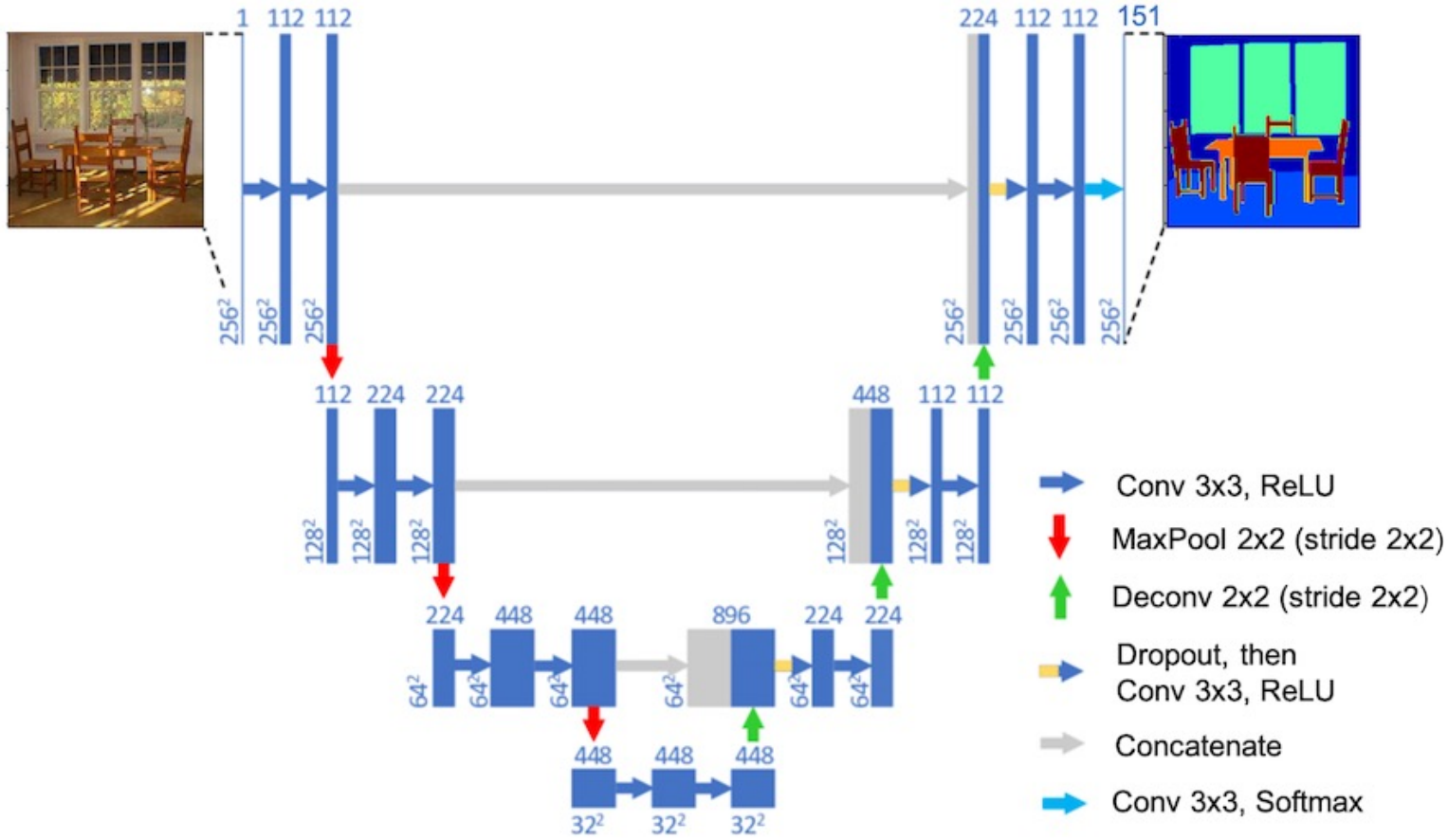


# U-Nets




Nikolas Adaloglou

# U-Nets

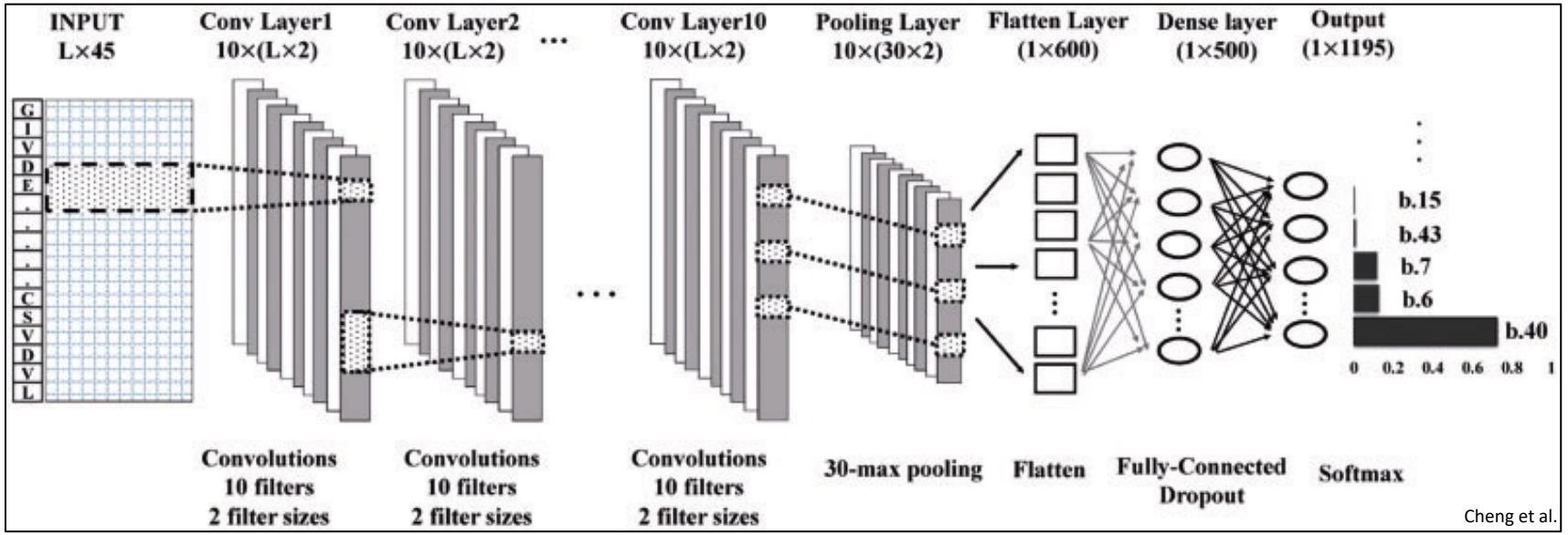



All padding is “same”. All strides 1x1 unless specified. Dropout rate 0.2

Silburt, et al.

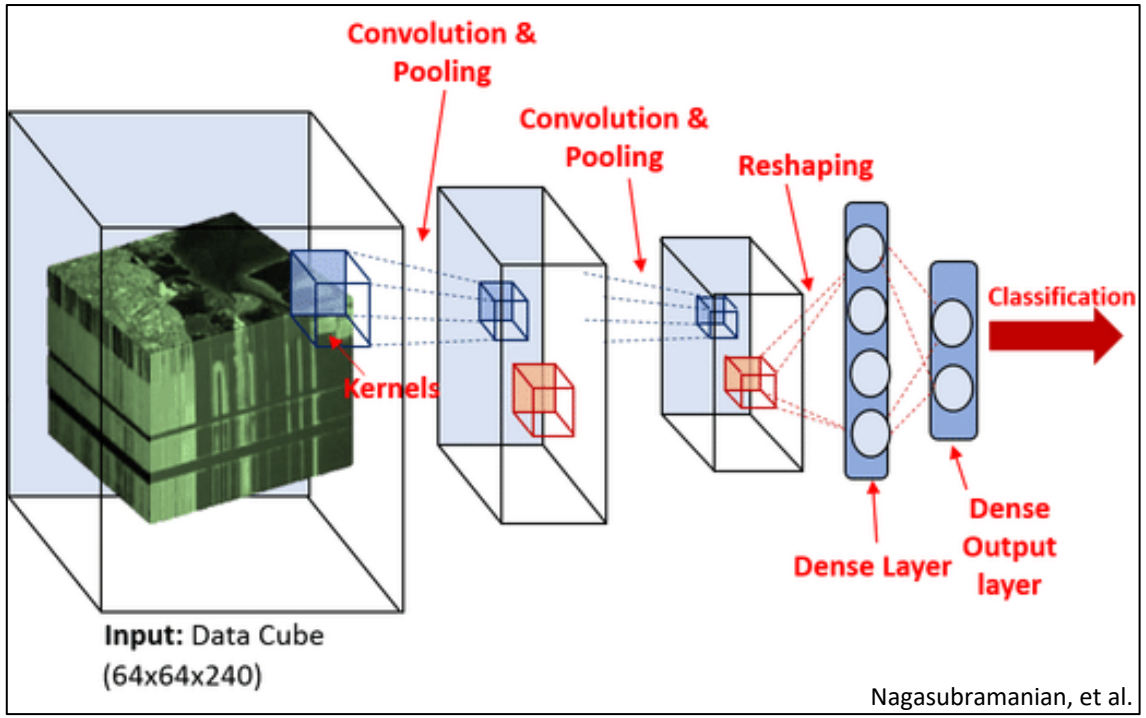
# 1D & 3D CNNs

- Convolutional networks are not limited to 2D images
- Convolution works on **1D** data with **1D** filters
  - E.g. audio recordings, Internet traffic rates, neuron spike trains



# 1D & 3D CNNs

- Convolutional networks are not limited to 2D images
- Convolution works on **3D** data with **3D** filters
  - E.g. video, volumetric imaging
- Keras `Conv1D` and `Conv3D` layers let you to build 1D and 3D CNNs



# Training CNNs in Keras

- See today's Jupyter notebook for CNN creation and training examples

- Keras comes with many existing architectures for easy* use
  - https://keras.io/api/applications/

* importing and compiling the architectures is easy…getting high performance on your data often isn't

# Questions?