CS195: Computer Vision

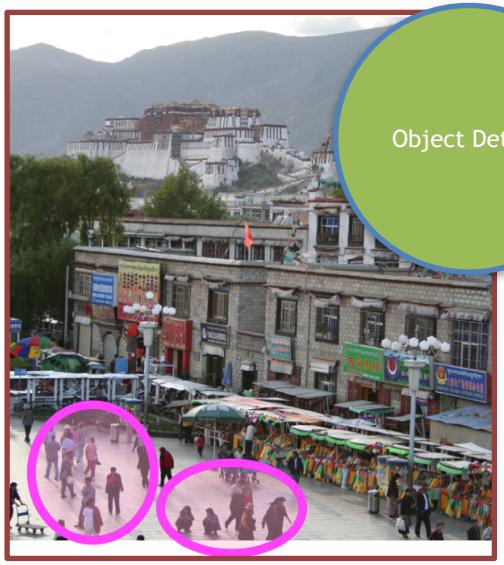
Object Recognition
Image Classification and Neural Networks (NN)
NN type: Convolutional Neural Network (CNN)

Wednesday, September 11th, 2024

Drake
UNIVERSITY

Object Recognition

What type of recognition problem?

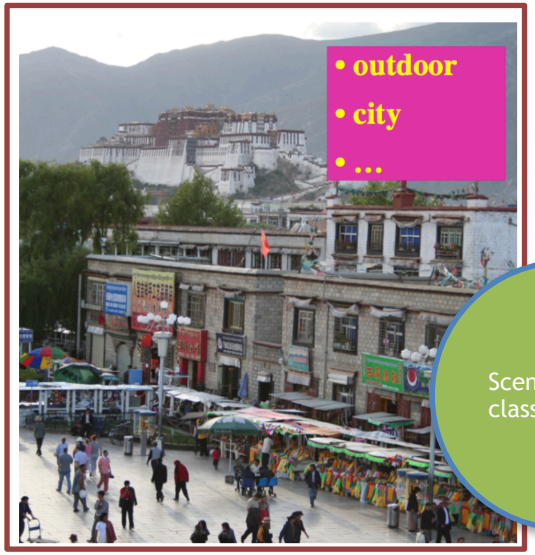


Object Detection

What type of recognition problem?

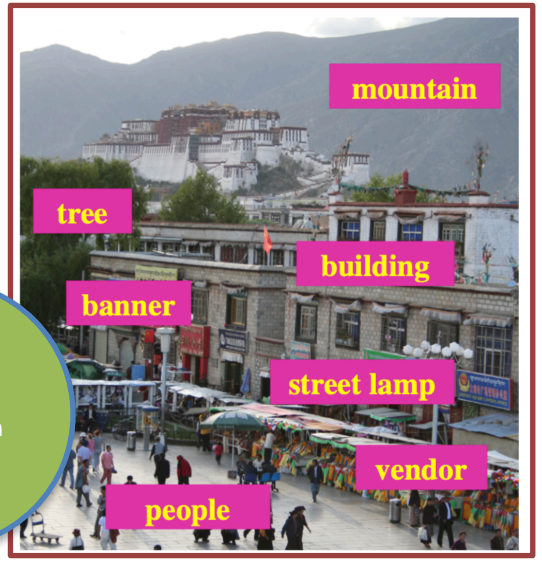
Single label classification

What type of recognition problem?



Scene and context classification

What type of recognition problem?



Multi-label classification

Object Recognition

What type of recognition problem?

Semantic segmentation

Current focus is on object classification using deep learning models

- Determine whether a given photo contains a ‘Dog’, ‘Cat’, ‘Horse’, or another specific object.

Current focus is on object classification using deep learning models

- Various types of **deep learning model based** — more specifically neural networks — can be employed to categorize an image into distinct classes

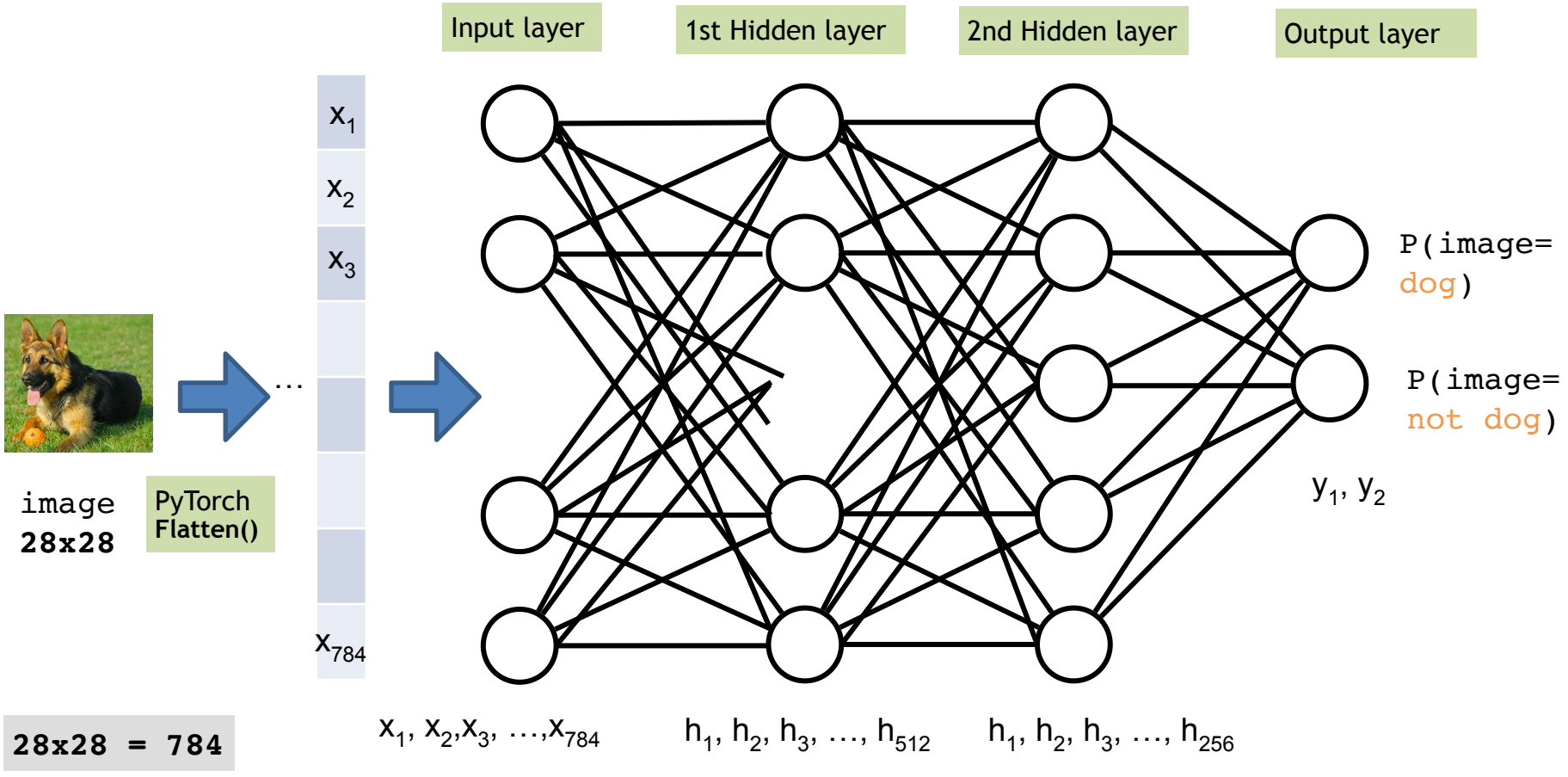
    - **Multilayer Perceptrons (MLP):** is the simplest type of neural network. It consists of perceptrons (aka nodes, neurons) arranged in layers
    - **Convolution Neural Network (CNN):** good for computer vision (CV) tasks
    - **Transformers:** rising star DL model; it had its inception in Natural Language Processing domain but is now gradually taking over all other AI domains such as Computer Vision, Audio/Speech, Robotics
    - **Very Recent additions (Early 2024):**
        - Mamba Network
        - Kolmogorov Arnold Networks (KAN)

Today's Agenda

- **Convolutional Neural Network (CNN):** good for computer vision (CV) tasks
  - Convolution operation
  - Nonlinearity
  - Pooling operation
  - CNN: convolutional layer + nonlinearity + pooling layer

Recall: Multilayer Perceptron (MLP)

- A **multilayer perceptron** is the simplest type of neural network. It consists of perceptrons (aka nodes, neurons) arranged in layers

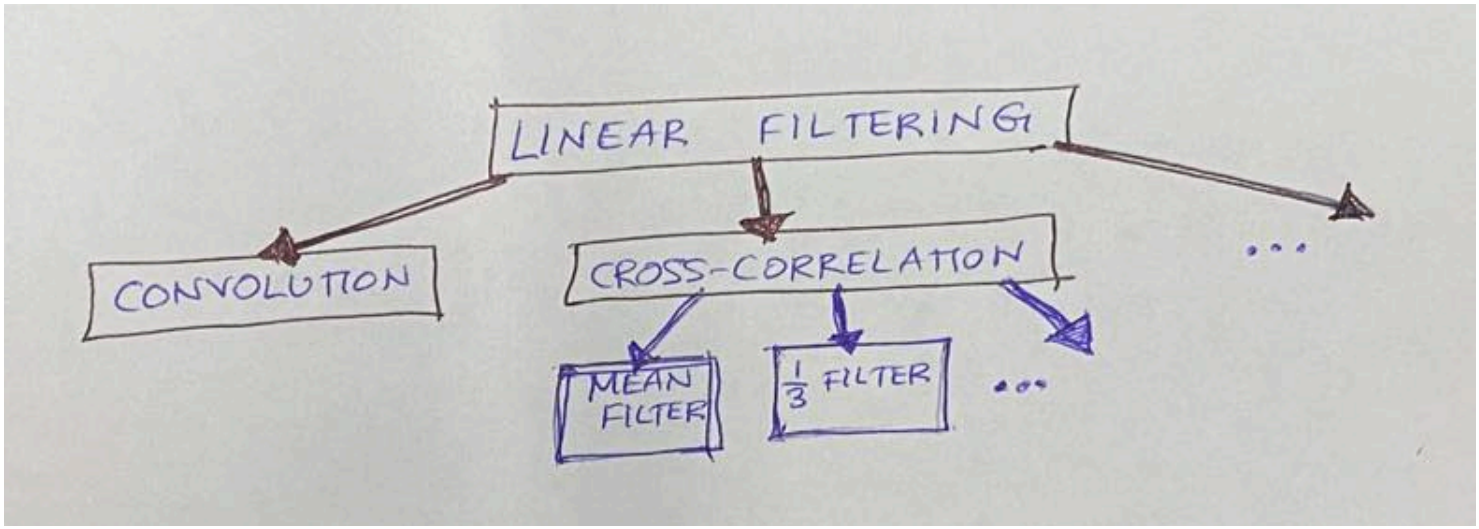


Today's Agenda

- **Convolutional Neural Network (CNN):** good for computer vision (CV) tasks
  - Convolution operation
  - Nonlinearity
  - Pooling operation
  - CNN: convolutional layer + nonlinearity + pooling layer

Convolution Operation

- **Convolution operation** falls within a more general form operation call **linear-filtering**
  - replace each pixel by a linear-combination of its neighbors

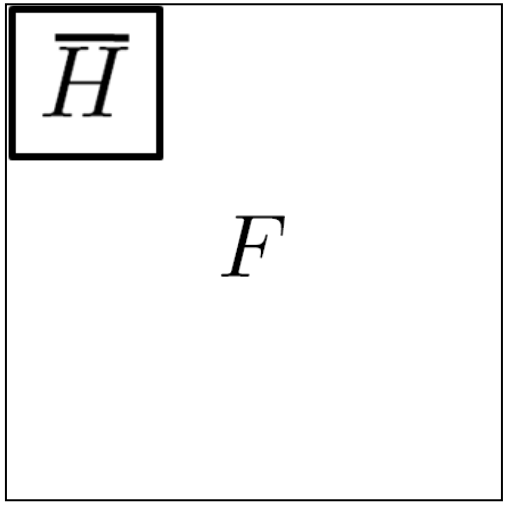


Convolution Operation

- What does a **convolution operation** do?

- In an ideal **convolution operation,** a kernel is “flipped” (horizontally and vertically) and then it is applied through the image (from left to right, and top to bottom)

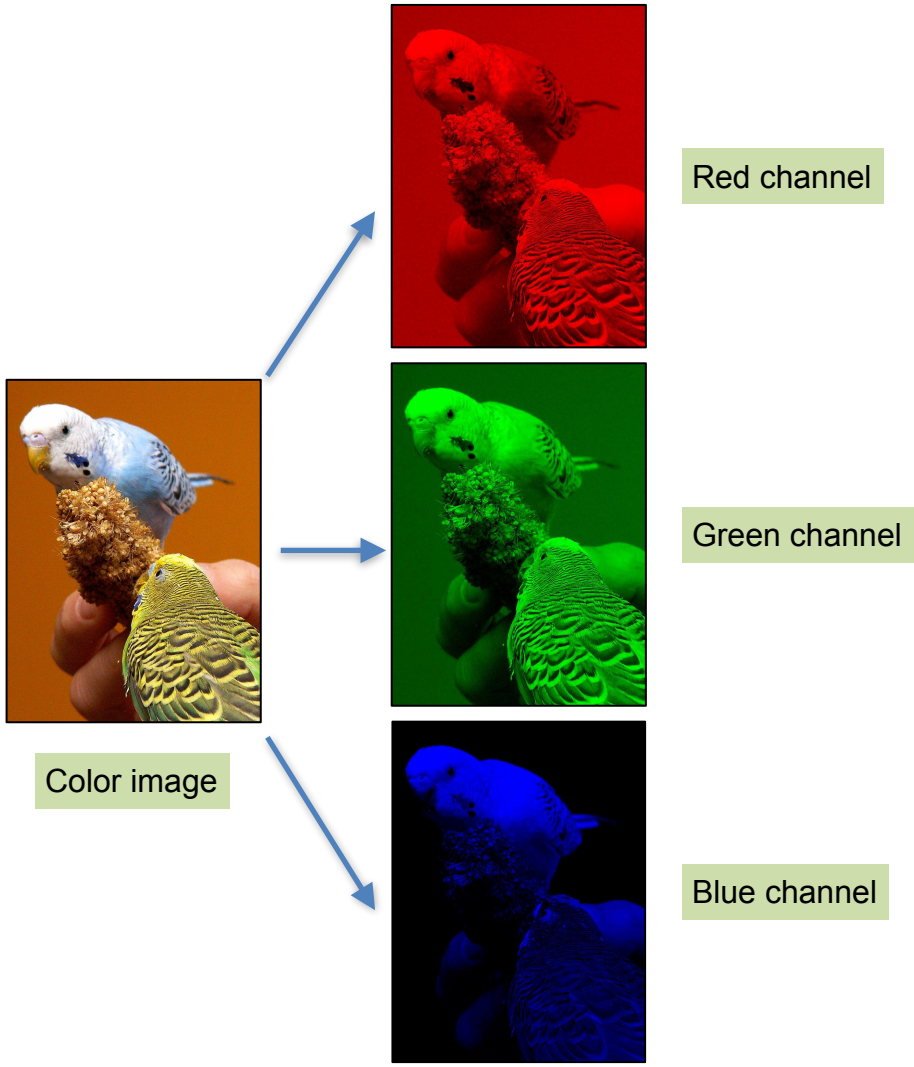
kernel of size
3x3 or 5x5



Convolutional Neural Network (CNN)

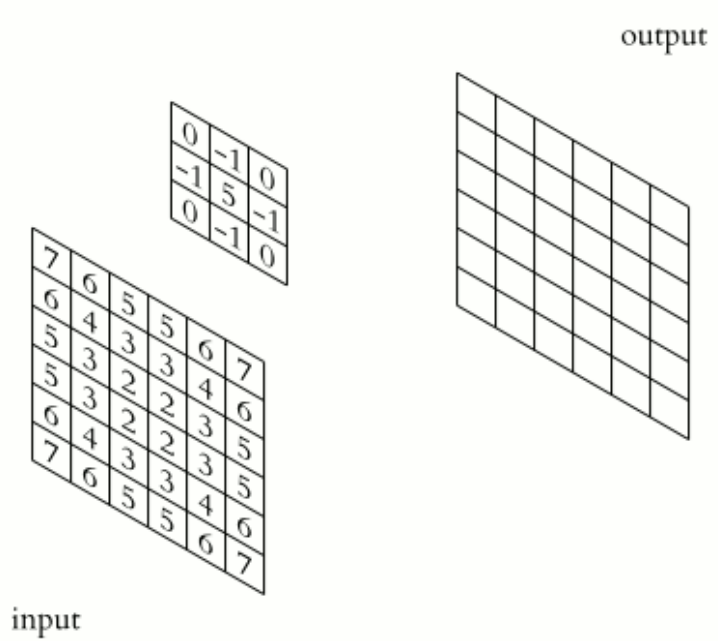
- A **convolutional neural network** that applies **convolutional filters** on grid-like input such as a image

- Image data is represented as a two-dimensional grid of pixels, either grayscale (monochromatic) or color (RBG)
  - each pixel corresponds to one or multiple numeric values: if it's grayscale, it is one number, if it's color, it corresponds to 3 numbers (a red, a green and a blue value)



Convolution Operation

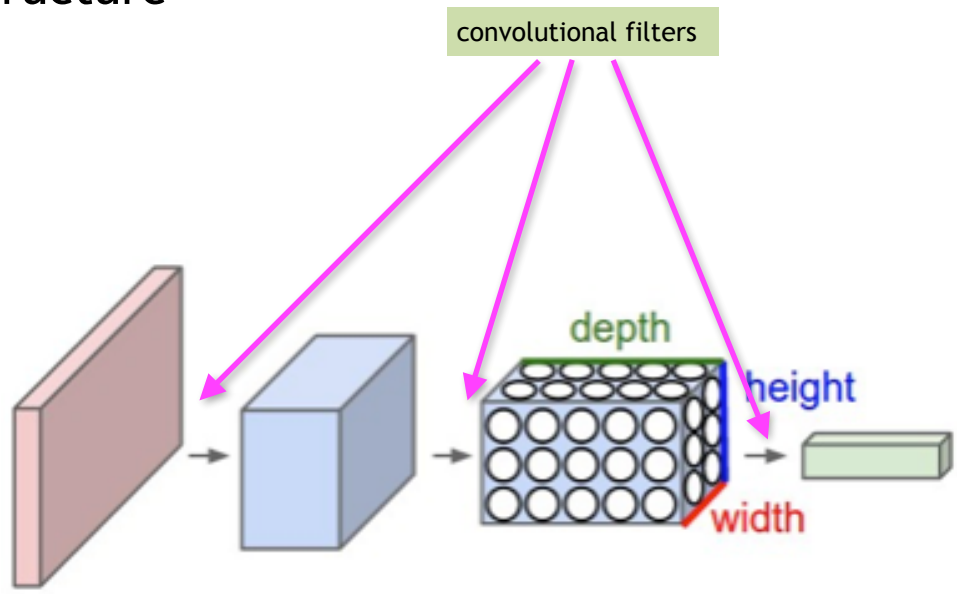
- What does a **convolution operation** do?
- **convolution operation** can be achieved with a series of dot products between portions of input feature map and a convolution filter (kernel) weights



visualization shows a convolution filter applied to an image, resulting in the convolved feature

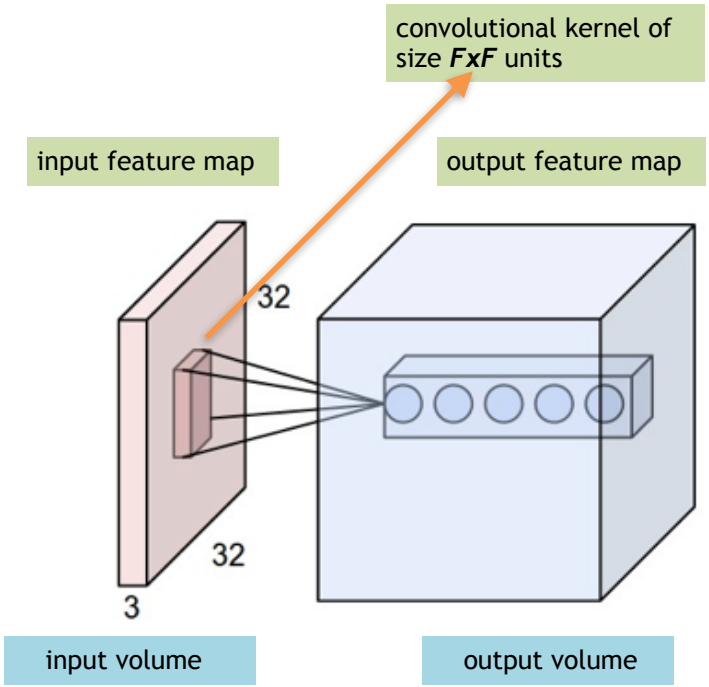
Convolutional Neural Network (CNN)

- A **convolutional neural network (CNN)** is a neural network with specialized connectivity structure



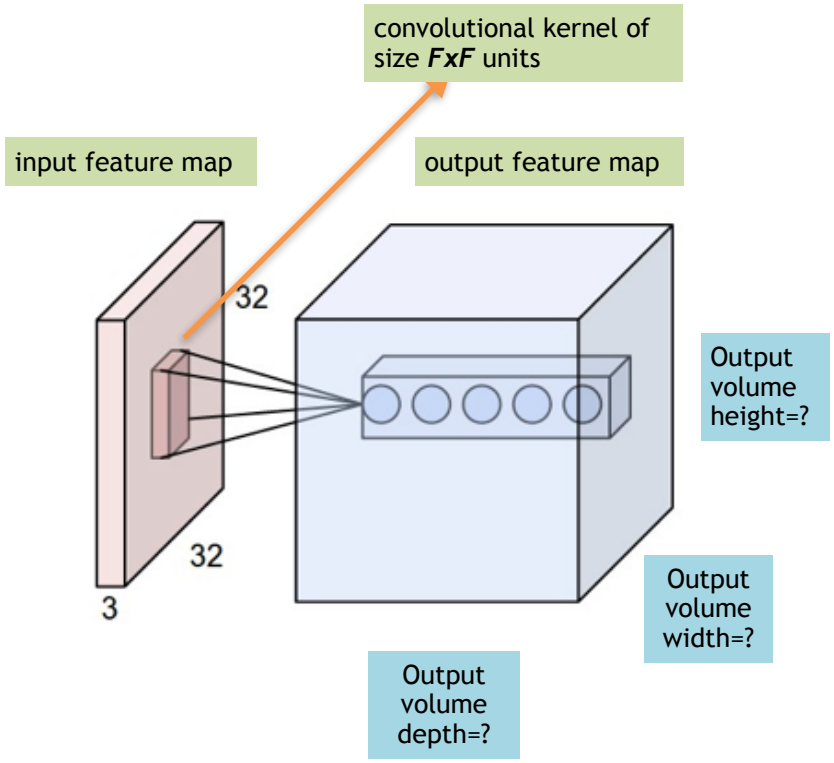
- Every layer of a CNN transforms the input volume to an output volume of neuron activations. The red input layer holds the image, so its width and height would be the dimensions of the image, and the depth would be 3 (Red, Green, Blue channels)

Convolutional Neural Network (CNN)

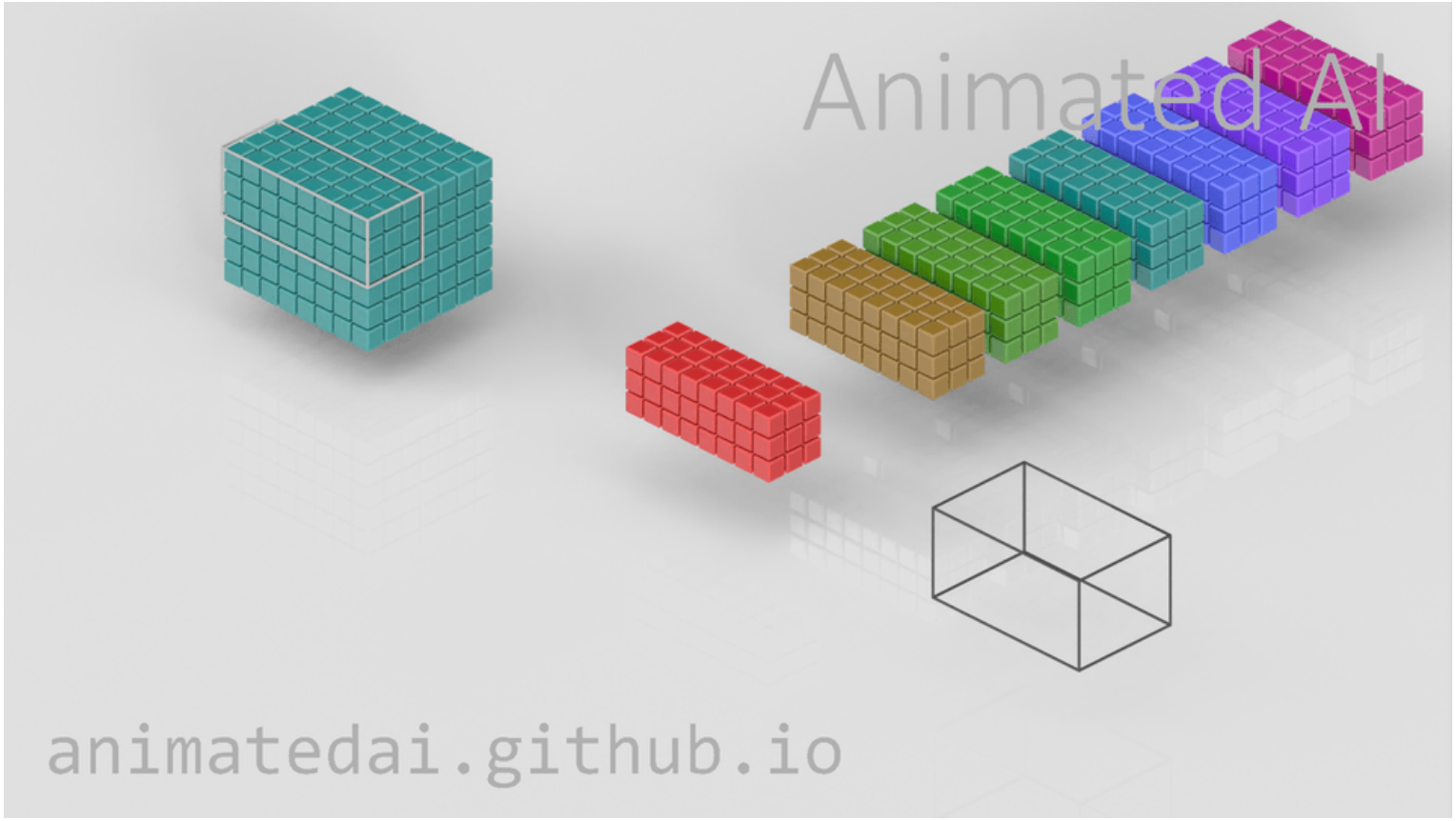


- Weights correspond to the filter (kernel) values
- **Convolutional neural network** can **learn** their own filters!
  - We do not need to provide the values inside the kernel

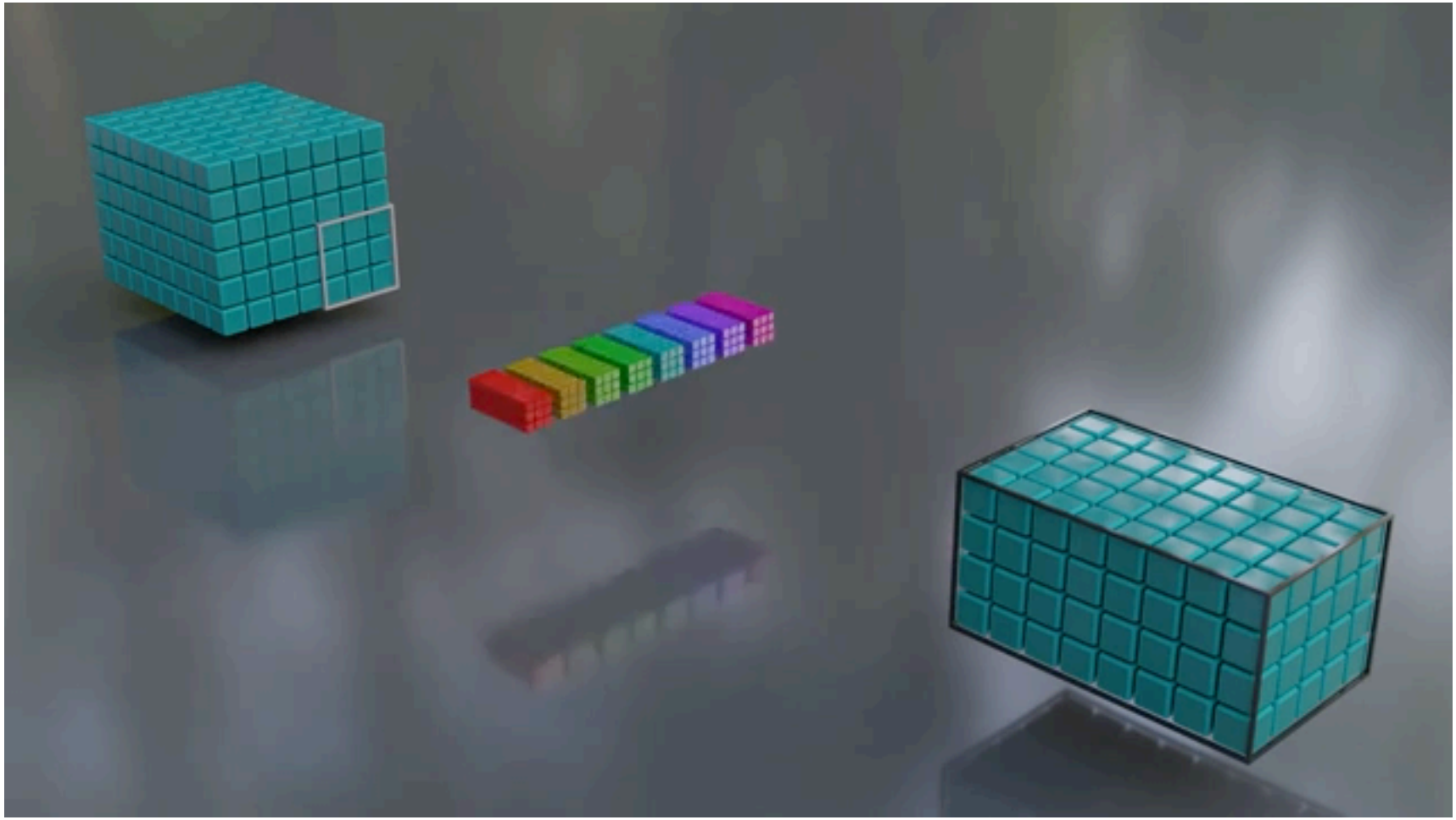
CNN: How to calculate the output volume size?



CNN: How to calculate the output volume size?



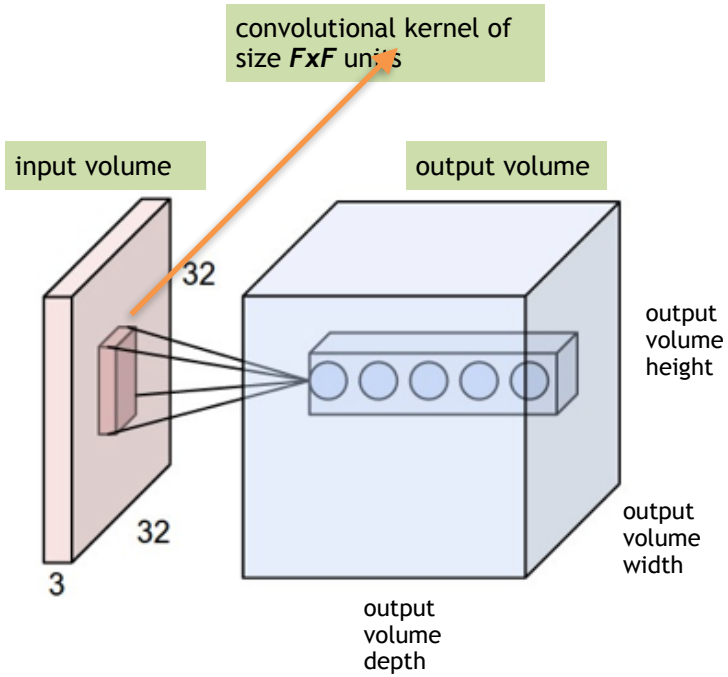
CNN: How to calculate the output volume size?



https://www.youtube.com/watch?v=w4kNHKcBGzA&t=210s

CNN: How to calculate the output volume size?

- An input volume has size ***(WxWx3)***, eg, (227, 227, 3)
- Filter size/receptive field is ***(FxF)***, eg, (11x11)
- Spatial Stride ***S***, eg, ***S***=4
- Padding size ***P***, eg, ***P***=0
- Number of filters ***K***, eg, ***K***=96

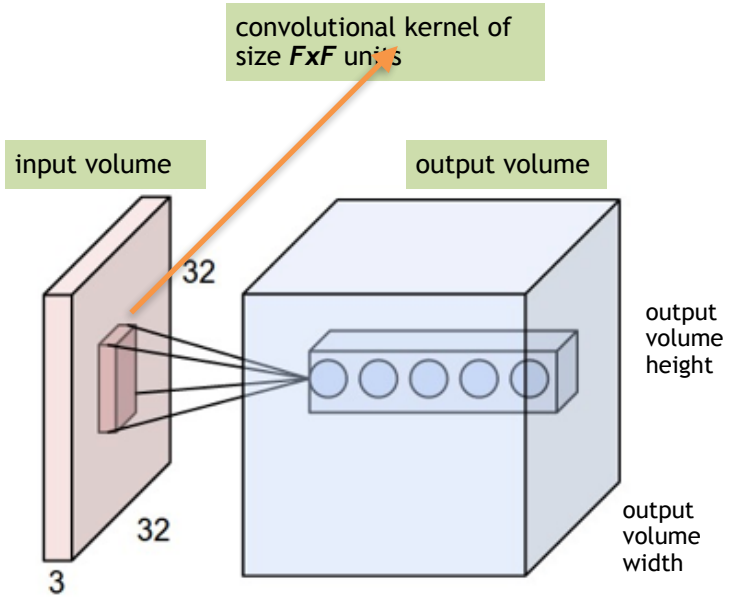


- Size of the output volume width and output volume height as a function of ***W***, ***F***, ***S***,and ***P*** as follows:

$$\text{output volume width/height} = \frac{(W - F + 2P)}{S} + 1 = \frac{(227 - 11 + 2*0)}{4} + 1 = 54+1 = 55$$

How to calculate the output volume size?

- An input volume has size **(W_1 x H_1 x D_1)**
  - Filter size/receptive field is ***(FxF)***
  - Spatial stride size ***S***
  - Padding size ***P***
  - Number of filters ***K***



- Spatial sizes of the output volume ***(W_2 x H_2 x D_2)***

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

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

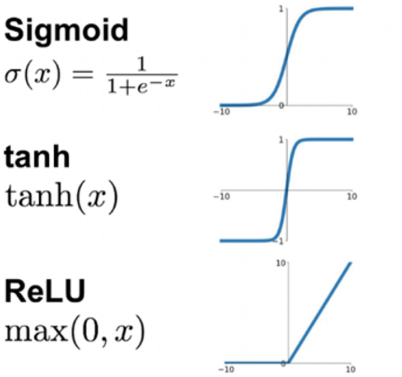
$$D_2 = K$$

- Number of filter weight parameters ***= (F x F x D_1) x K***
- Number of bias parameters = ***K***

Today's Agenda

- **Convolutional Neural Network (CNN):** good for computer vision (CV) tasks
  - Convolution operation
  - Nonlinearity
  - Pooling operation
  - CNN: convolutional layer + nonlinearity + pooling layer

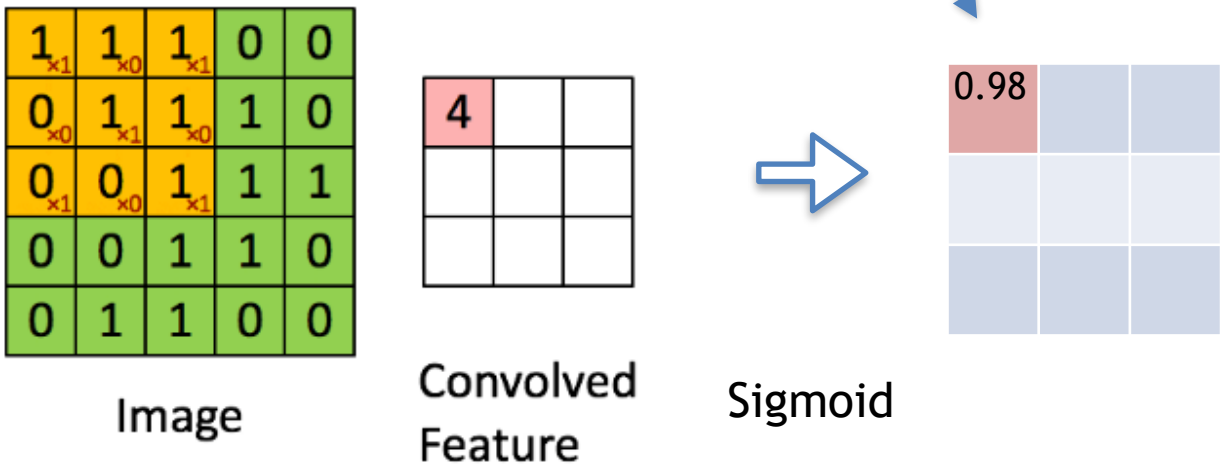
Nonlinear Function



- Just like an MLP, each convolutional output goes through a non-linear function such as Sigmoid, Tanh, or Rectified Linear Unit (ReLU)

$$convolution = 1*1+1*0+1*1+0*0+1*1+1*0+0*1+0*0+1*1 = 4$$

$$Sigmoid(4) = \frac{1}{1+exp(-4)} = 0.98$$



Today's Agenda

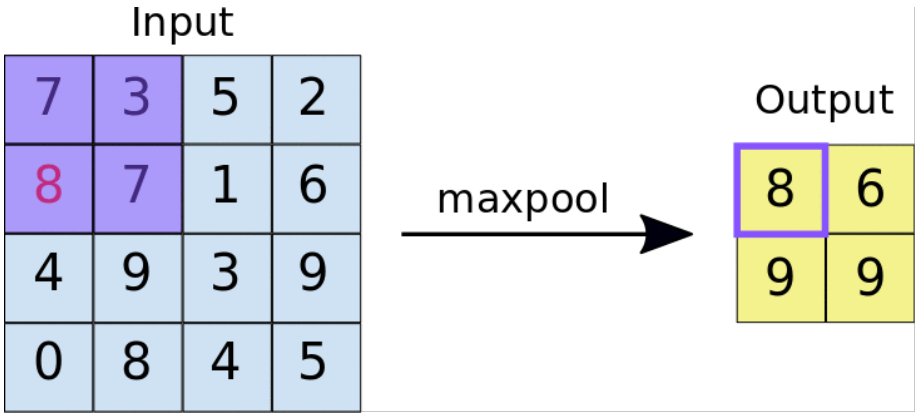
- **Convolutional Neural Network (CNN):** good for computer vision (CV) tasks
  - Convolution operation
  - Nonlinearity
  - Pooling operation
  - CNN: convolutional layer + nonlinearity + pooling layer

Pooling Operation

- Image data can get computationally inefficient, really quickly. To avoid this, we often toss in a layer that helps us to **summarize** and **downsample** the data

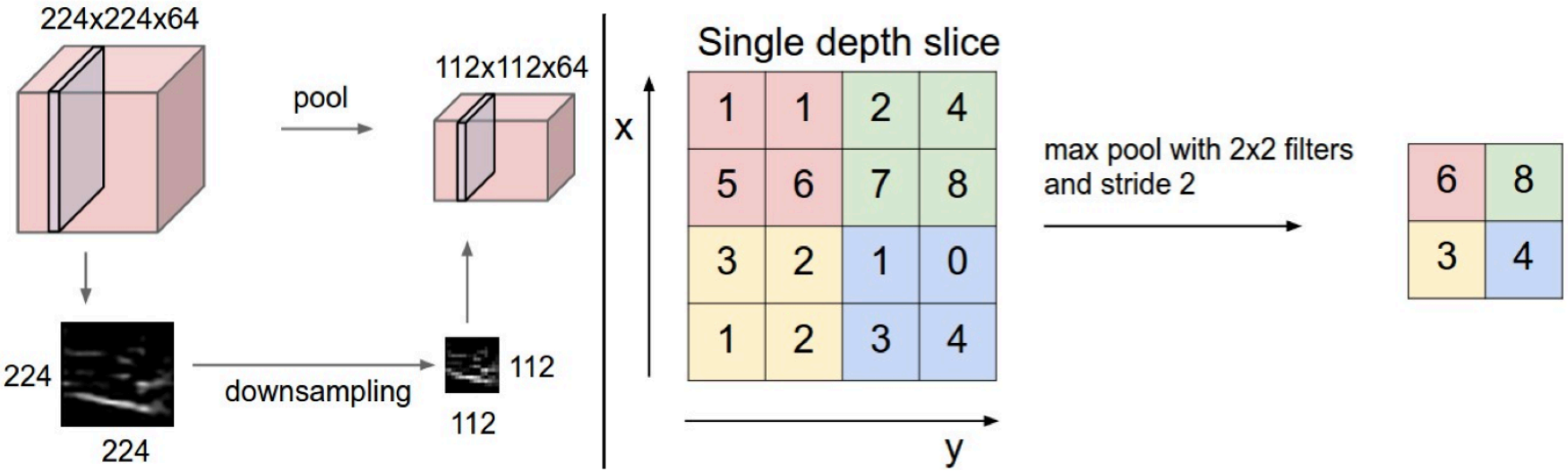
- In classical CNN, we find another useful operation called **pooling operation**

- A common pooling operation is **max pooling**, and its goal is to locally summarize the convolution. It performs something like a convolution, but rather than taking the dot product, it takes the maximum element in the filter area



Pooling Operation

- Pooling operation downsamples the volume spatially, i**ndependently in each depth slice of the input volume**
- Besides max pooling, other pooling operations include: **sum pooling**, average pooling



How to Compute the Output Volume Size for Pooling Operation?

- An input volume has size $(W_1 \times H_1 \times D_1)$
  - Filter size/receptive field is *(FxF)*
  - Spatial stride size ***S***
  - Padding size ***P***

- Spatial sizes of the output volume $(W_2 \times H_2 \times D_2)$

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

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

$$D_2 = D_1$$

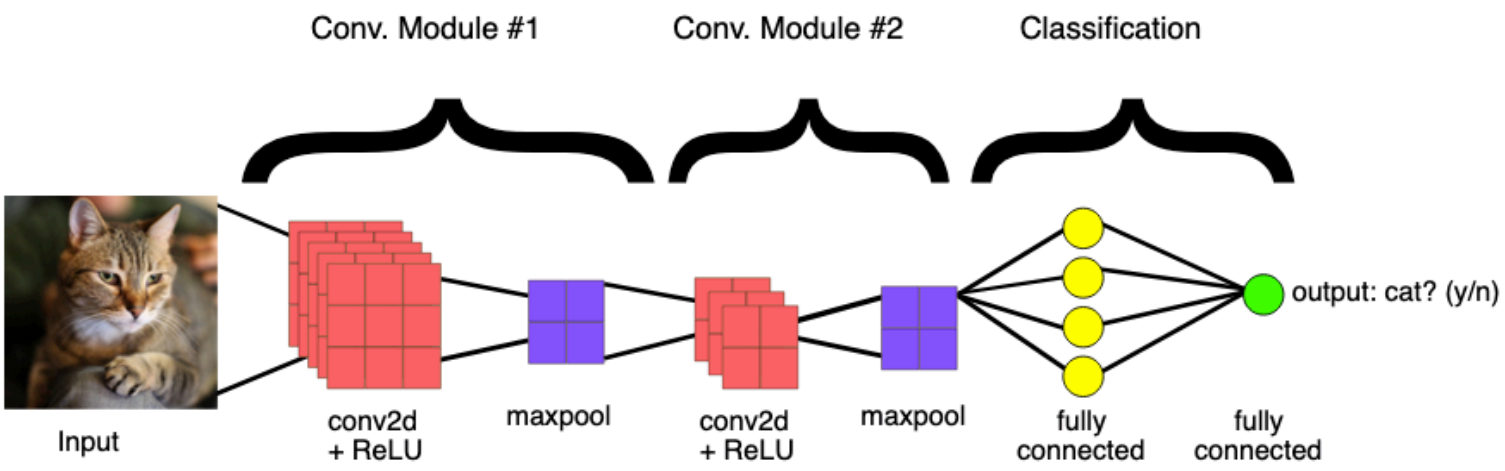
- Number of filter weight parameters = ***zero parameter***
  since it computes a fixed function eg max() or average()

Today's Agenda

- **Convolutional Neural Network (CNN):** good for computer vision (CV) tasks
  - Convolution operation
  - Nonlinearity
  - Pooling operation
  - CNN: convolutional layer + nonlinearity + pooling layer

CNN: A Composition of Convolutional Layers

- We've talked about **image data**, **convolutions**, **nonlinearity**, **max pooling**, and how they are related to some computer vision tasks. Let's connect the dots
  - input is an image (in this case a color image, so 3 channels—red, green, and blue)
  - there are several filters, not just one.
  - `Conv2D` layers with `ReLU` are often followed by `maxpool`
  - towards the end of the model, we switch to fully connected (`Dense`) layer
  - We have as many output nodes as we have classes to predict



Reference