Introduction to Convolutional Neural Networks (CNNs)

Jie Tang

Department of Computer Science & Technology

Tsinghua University

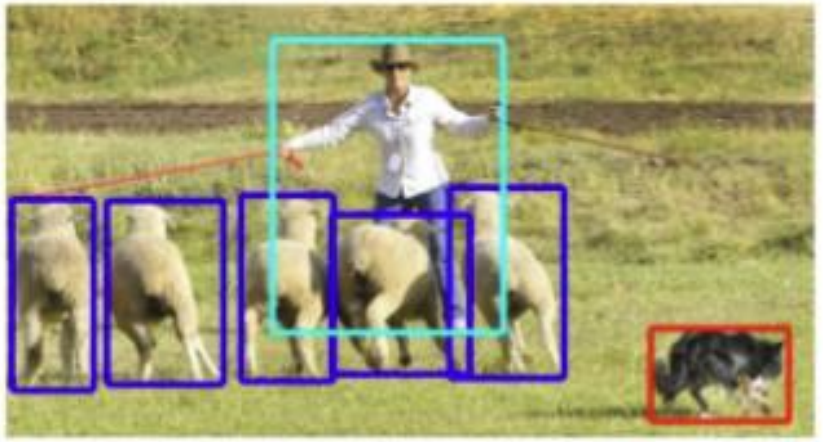
How do machines recognize our faces?

TMALL supermarket in Tsinghua campus

CV tasks

Image classification

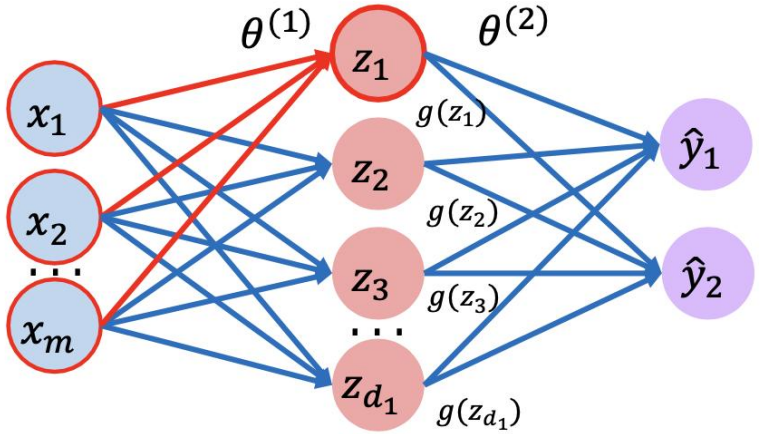
Object detection

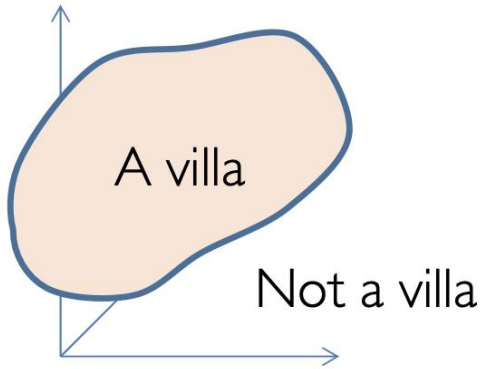


Semantic segmentation

Image generation

MLP for Vision ?





Dimensions
372x372

Input | Hidden Layer | Output

Result

Spatial Info lost

Input
N= 138,384

k hidden layers
Layer 1: α×138,384
Layer 2: β×138,384
Layer 3: γ×138,384

Output

Parameter explosion

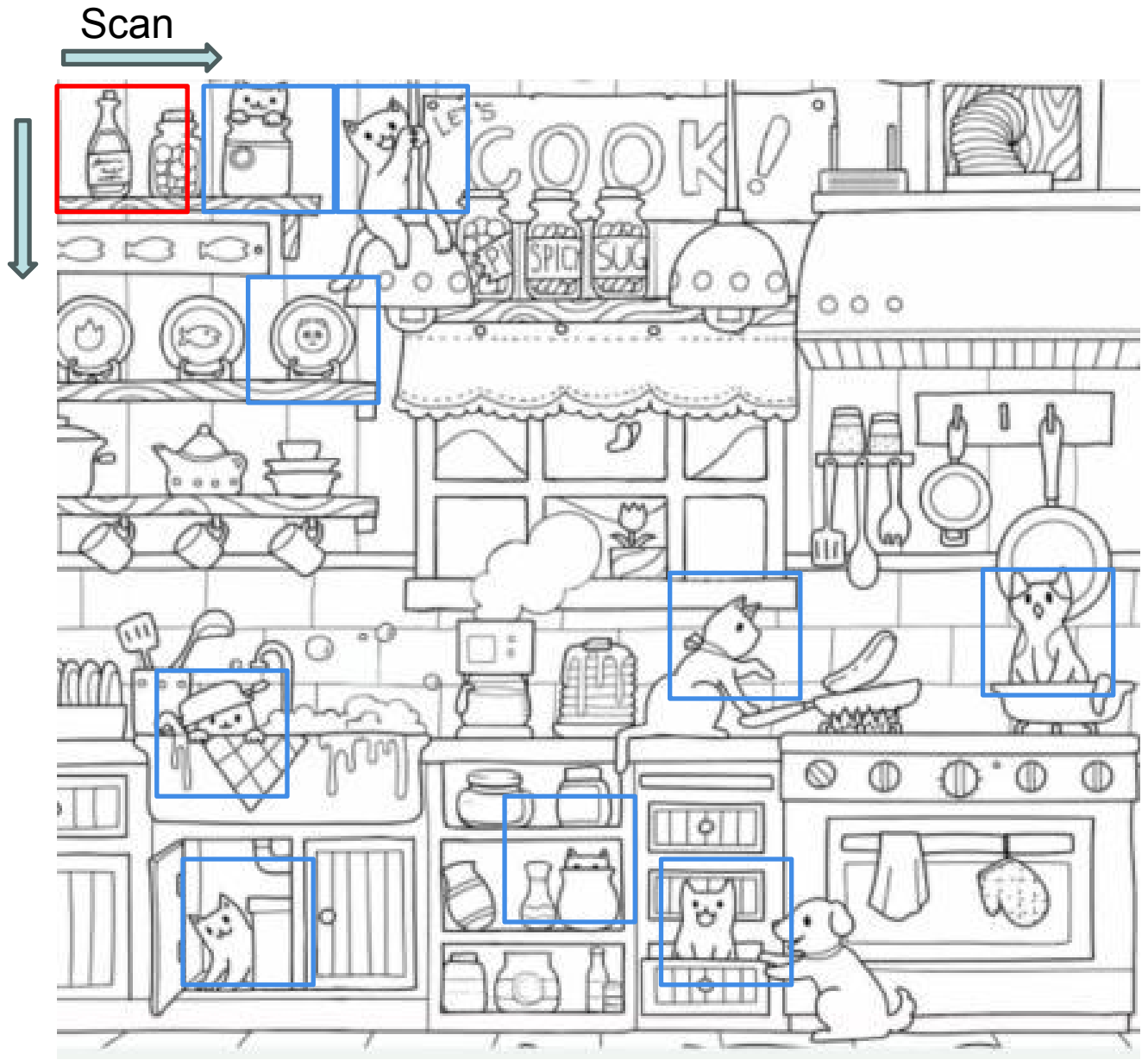
It is hard to adapt scenarios of viewpoint variation, scale variation, illumination conditions and so forth.

Object detection

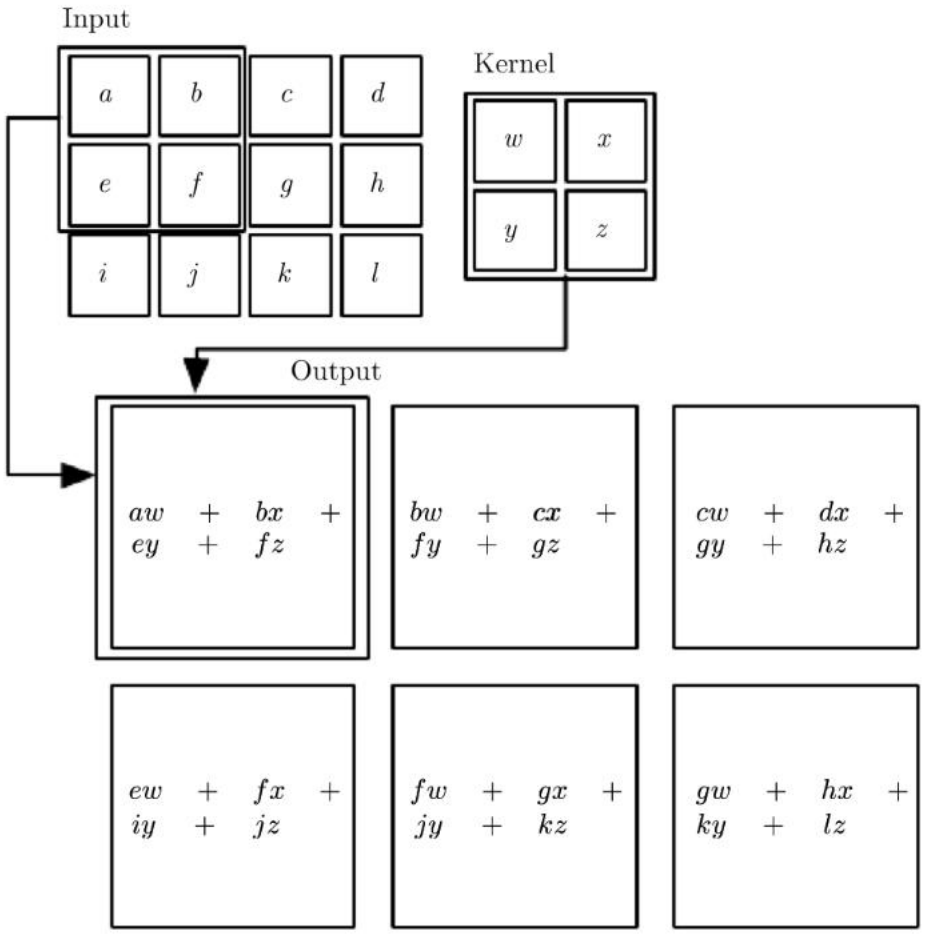
How many cats are in the picture?

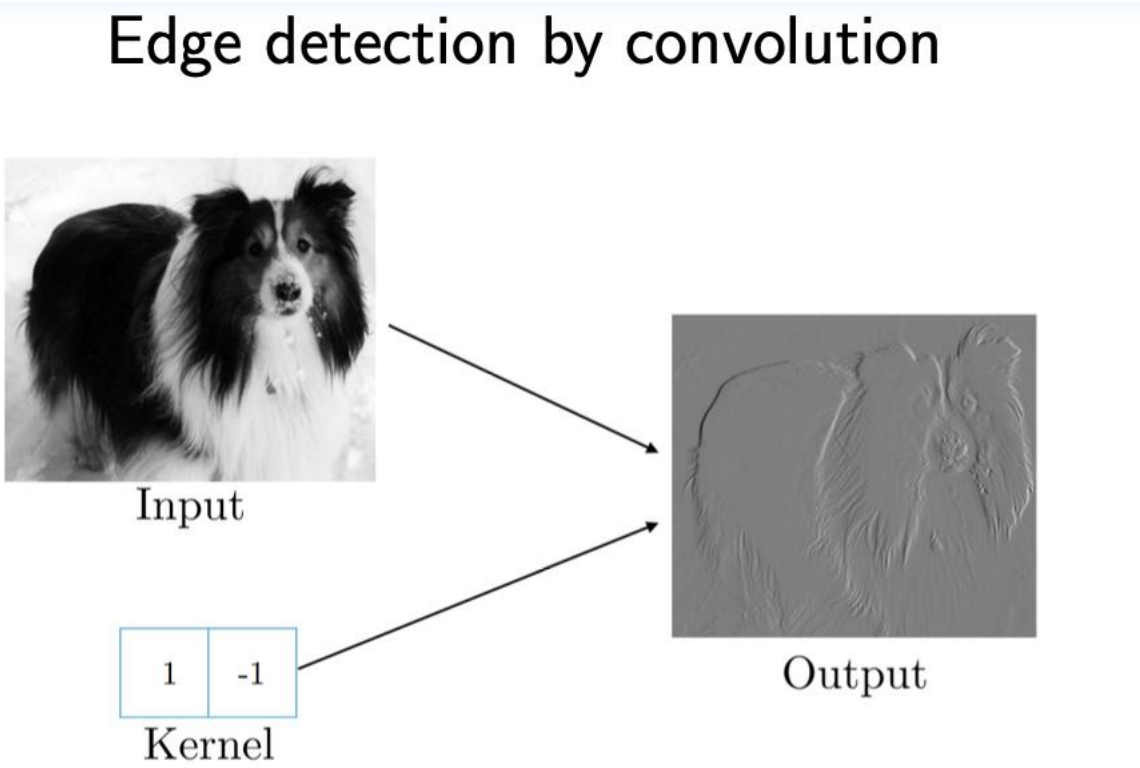
Object detection

How many cats are in the picture?

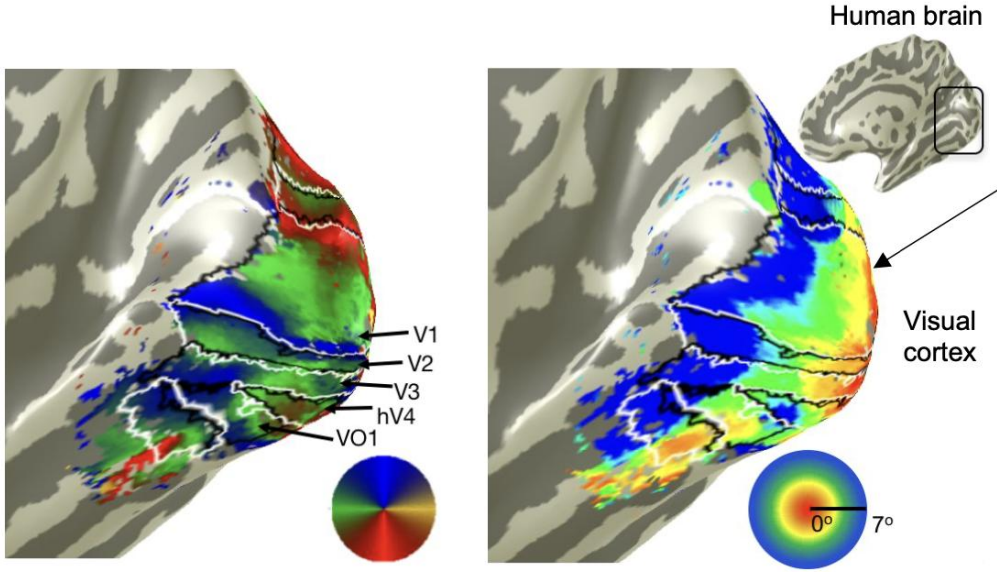


2D Convolution





Neuroscience on the Brain

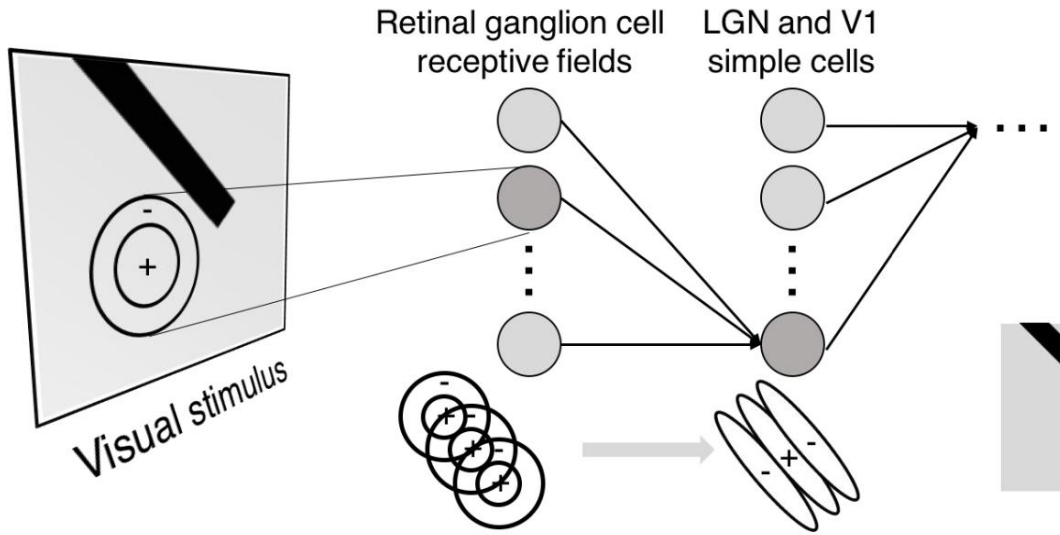


Topographical mapping in the cortex:
nearby cells in cortex represent nearby regions in the visual field

Simple cells:
response to *light orientation*

Complex cells:
response to light orientation and *movement*

Hypercomplex cells:
response to movement with and *end point*

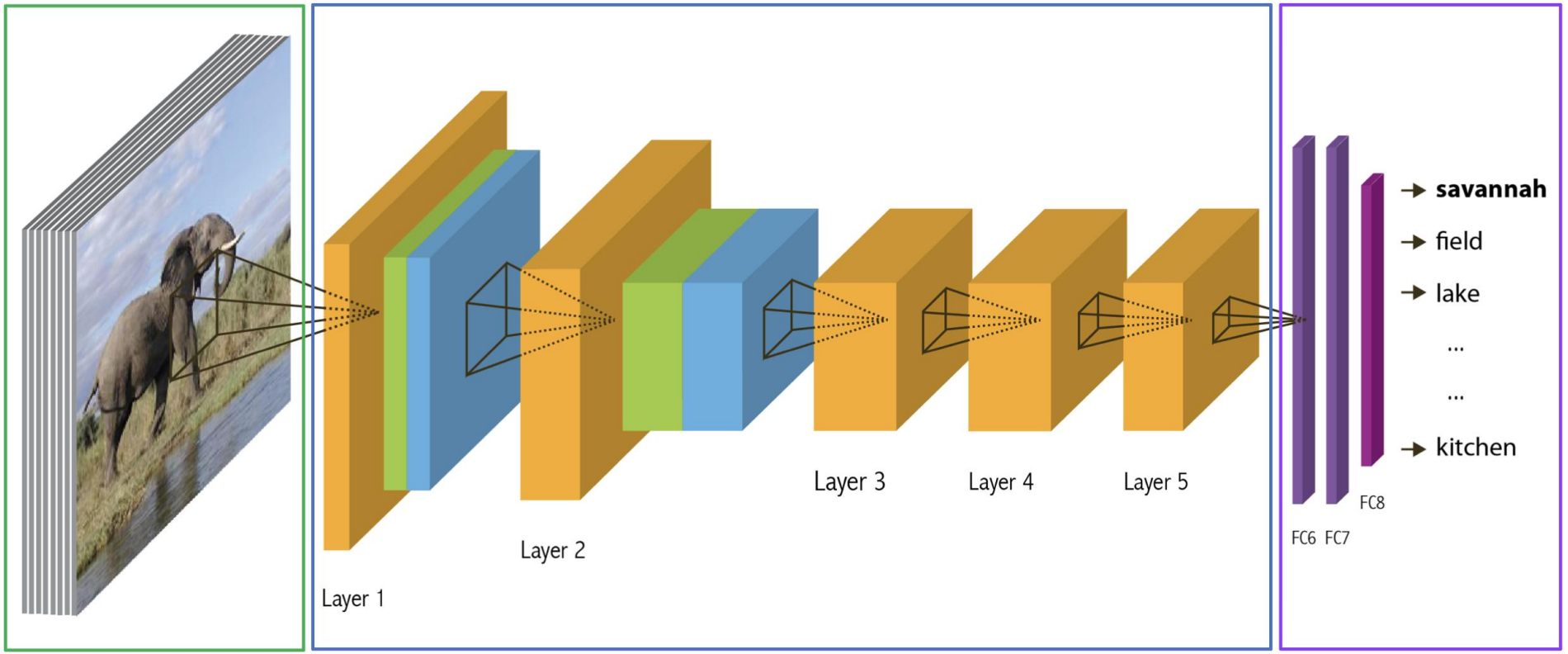


No Response

Response (end point)

Slide credit: CS231n

Convolutional Neural Network: Overview

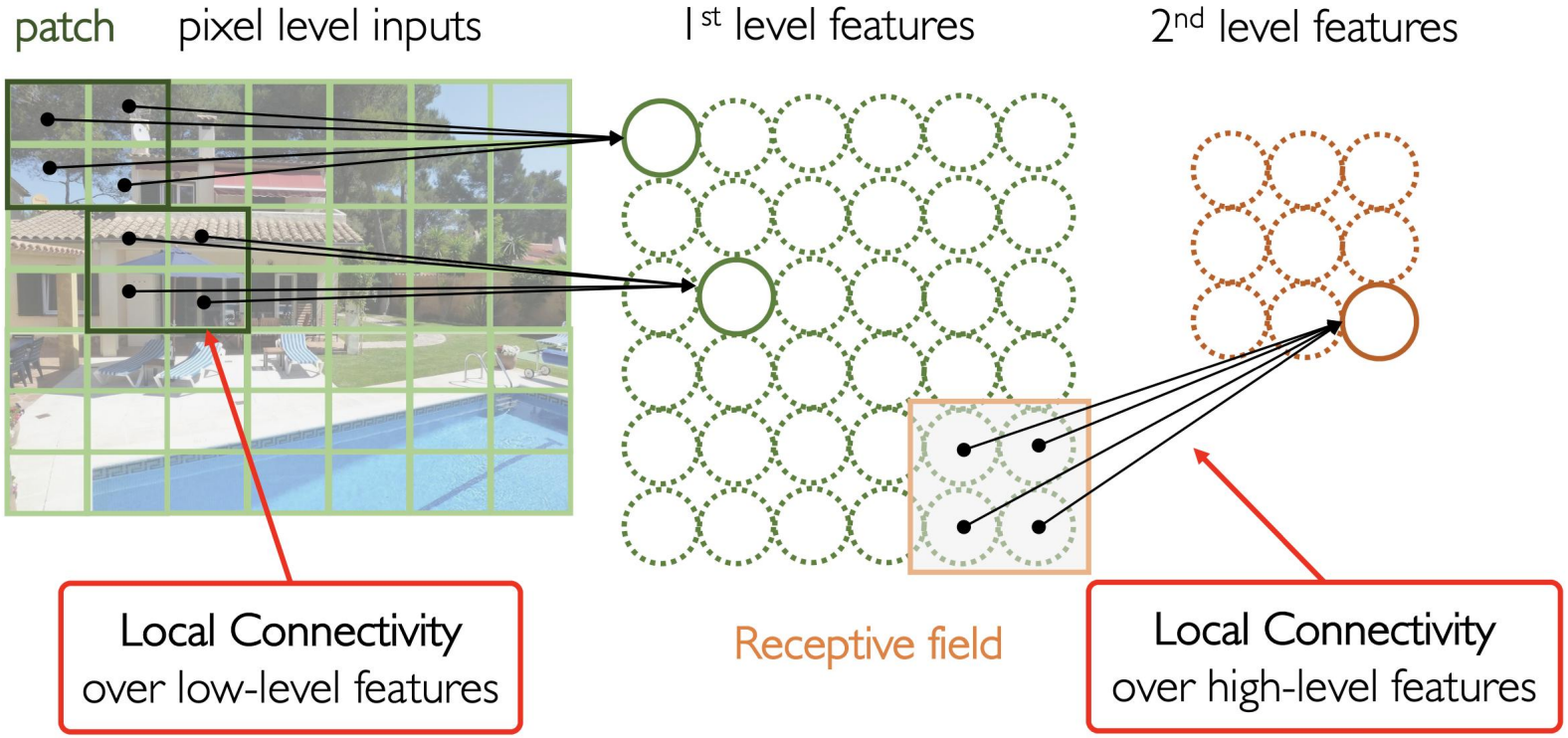


Scan the image

Generate hierarchy of features

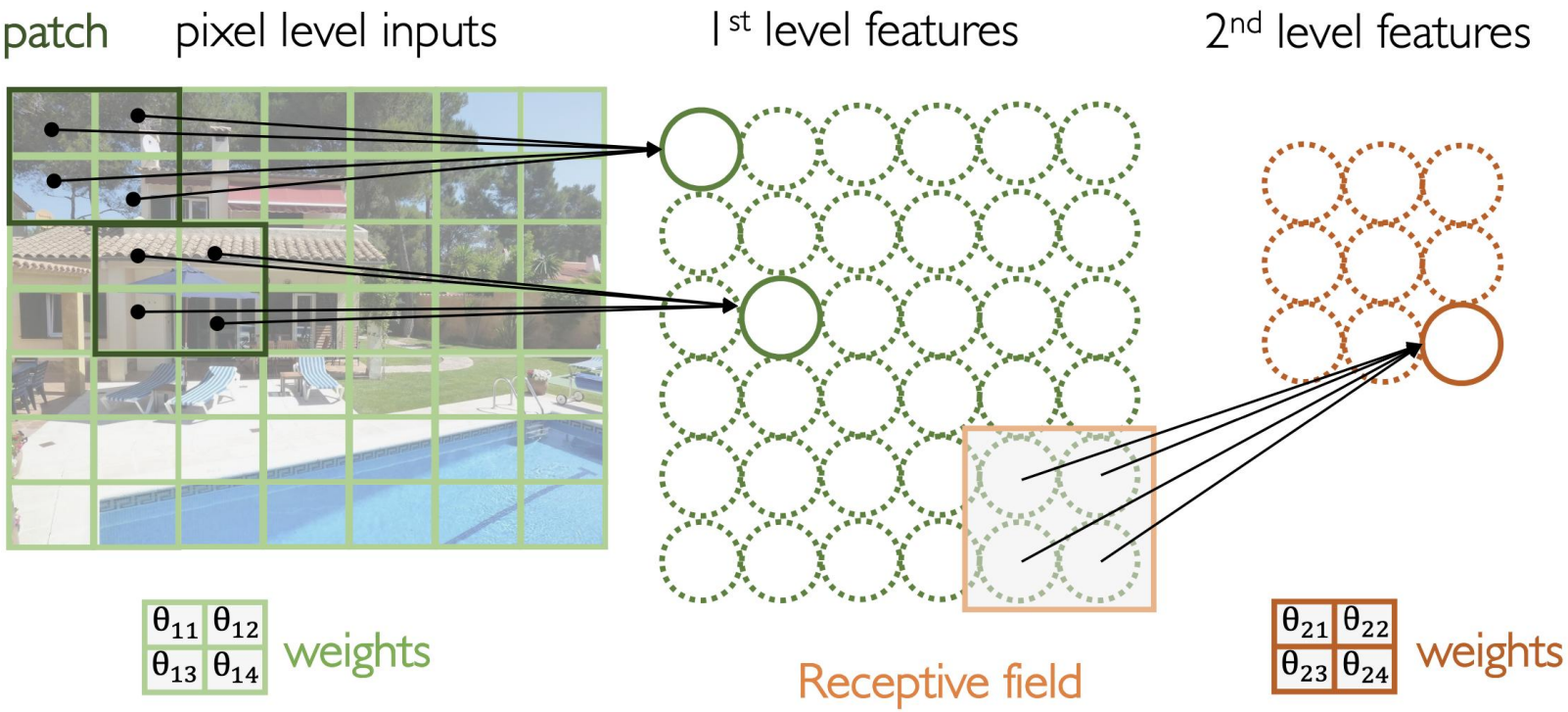
Recognize from high level features

Idea 1: Local connectivity



- Locality Assumption: local information is enough for recognition
- Connect each neuron to only a local region of the input volume

Idea 2: Parameter Sharing



- Shift Invariance Assumption: If a feature is useful at spatial position (x, y), then it should also be useful for other positions (x', y')
- Share the weights of sliding window over different spatial locations

What is Convolution?

- In mathematics, convolution is an operation on two functions (f and g)

$$\begin{aligned}(f * g)(t) &= \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \\ &= \int_{-\infty}^{\infty} f(t-\tau)g(\tau)d\tau\end{aligned} \tag{1}$$

- For functions f and g defined on the set Z of integers, we can define the discrete convolutions of f and g

$$\begin{aligned}(f * g)(n) &= \sum_{m=-\infty}^{\infty} f[m]g[n-m] \\ &= \sum_{-\infty}^{\infty} f[n-m]g[m]\end{aligned} \tag{2}$$

What is Convolution?

- Suppose we are estimating the location with a GPS. $f[n]$ is the value from GPS, but possibly with noise
- To obtain a less noisy estimation, we can use recent values of the GPS to smooth estimation with a weighting function $g[m]$, but of course recent value should have a higher weight

$$(f * g)(n) = \sum_{m=-\infty}^{\infty} f[m]g[n-m] \qquad (3)$$

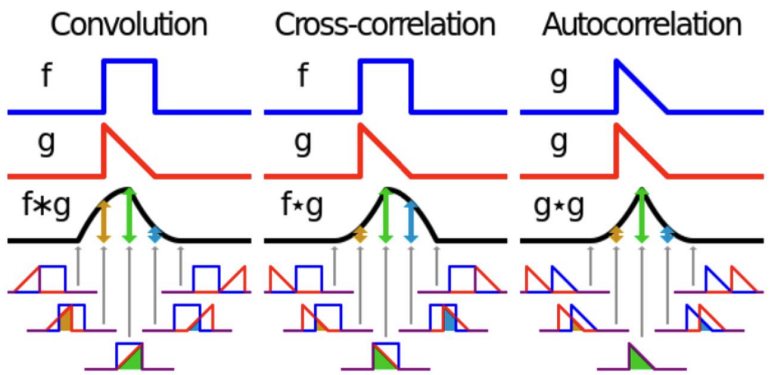
What is Convolution?

- In ML, the input usually forms a multi-dimensional structure. For example, convolution is used to transform a 2D input I with a 2D kernel K:

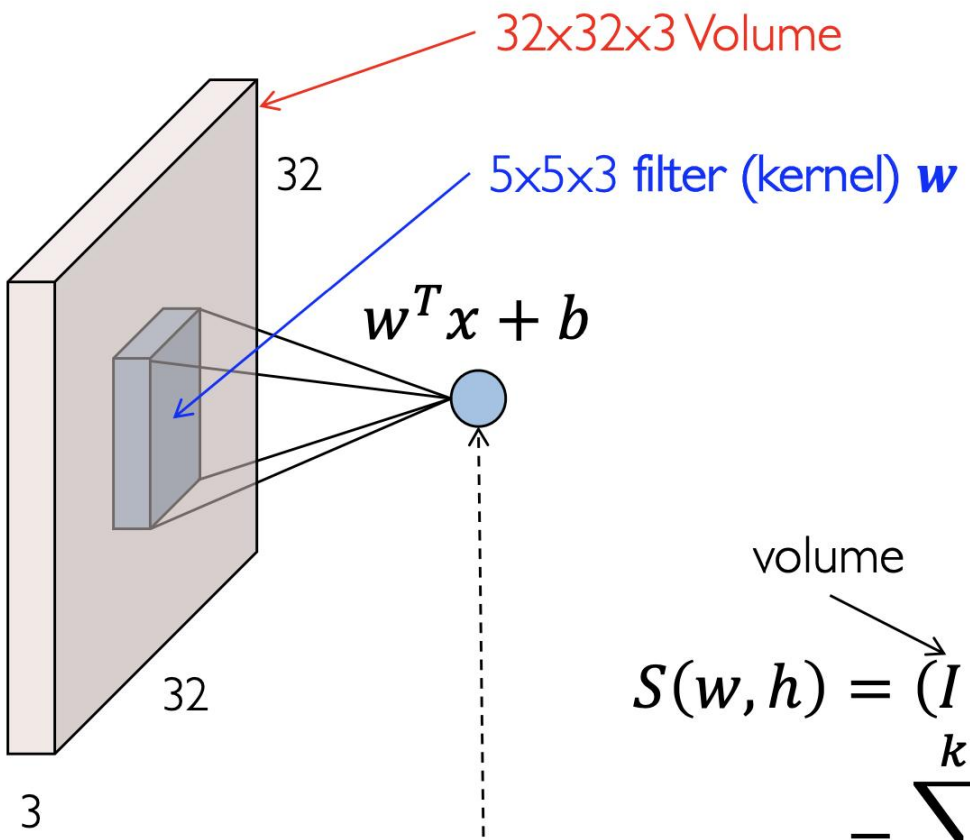
$$Z(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n)$$

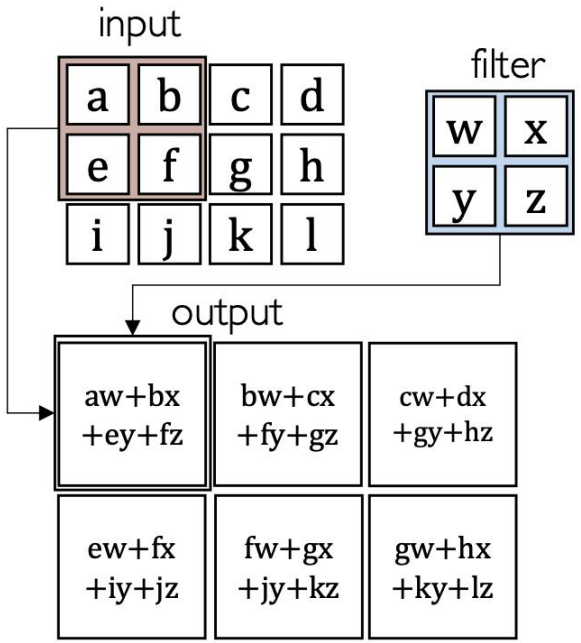
- In ML, learning algorithm based on convolution with kernel flipping will learn a kernel that is flipped relative to the kernel learned without flipping.
- So, many ML libraries implement cross-correlation but call it convolution:

$$Z(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n)K(m,n)$$



Convolution





volume filter

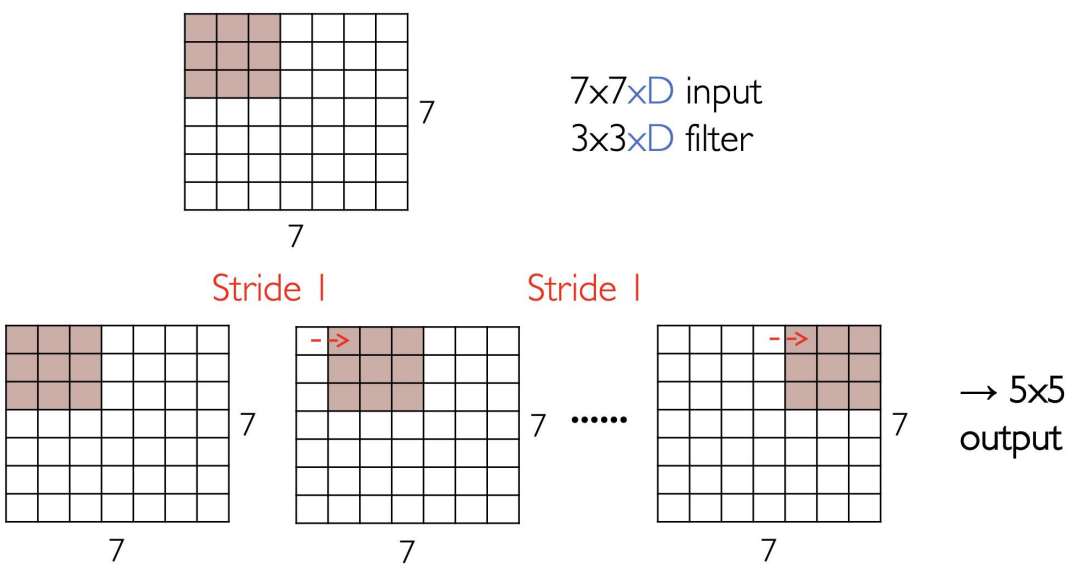
$$S(w,h) = (I \star K)(w,h)$$

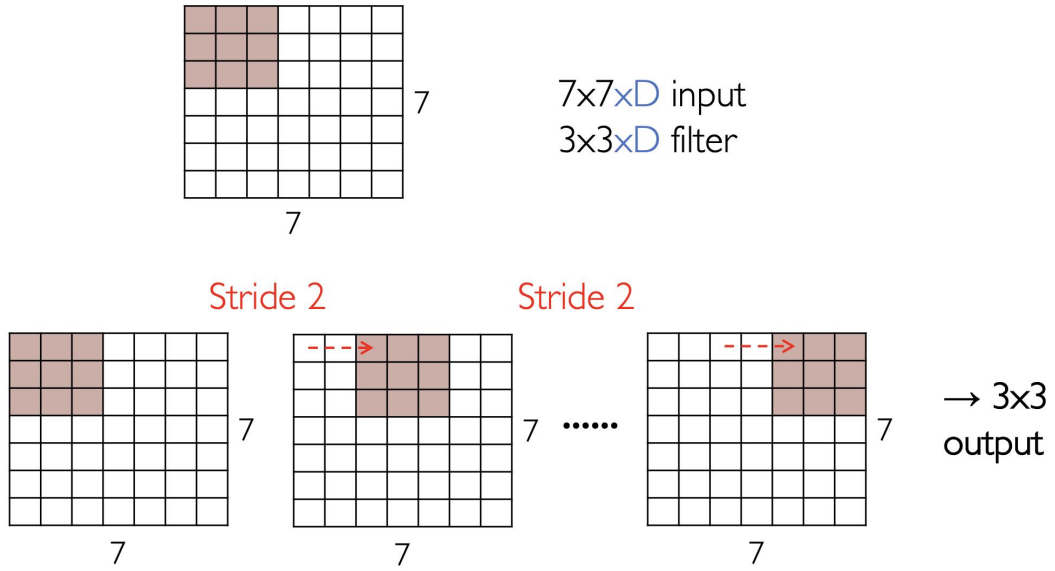
$$= \sum_{i=1}^{k} \sum_{j=1}^{k} I(w+i, h+j) K(i,j)$$

- The output of each neuron:
  - Dot product between filter and a small 5x5x3 chunk of the volume

Stride

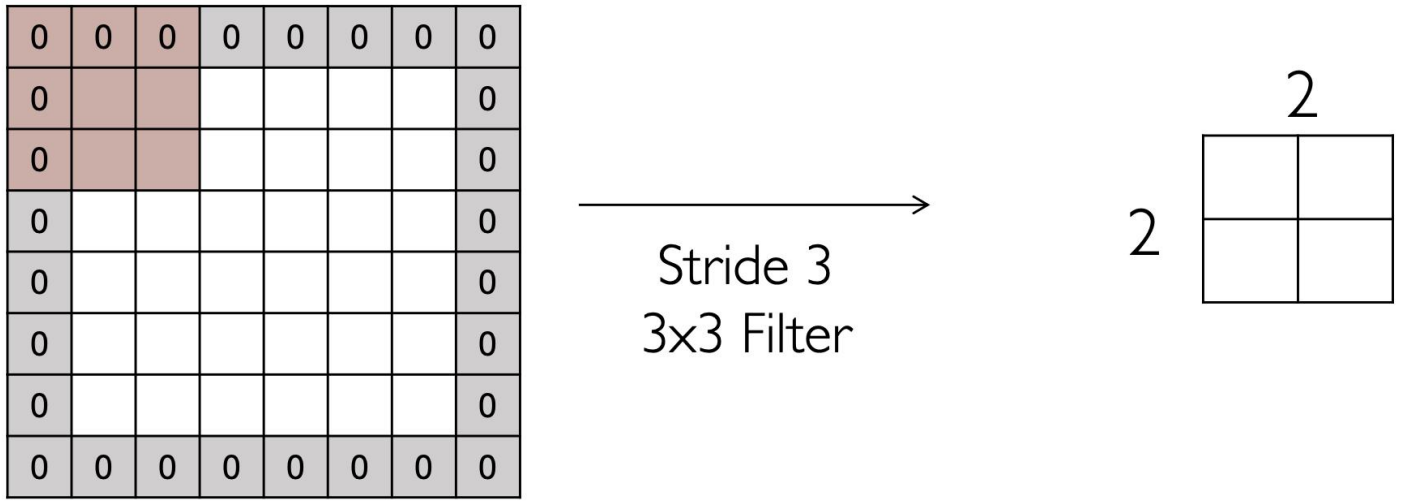
- In some cases we want to reduce the spatial resolution (subsample the output)
- Performed over the spatial dimensions for each channel





Padding

- May not fit? E.g. cannot apply 3x3 filter on 7x7 input with stride 3.
- Multiple layers of convolutions reduce the spatial sizes $W \times H$ and the information available at the boundary



(Pad P)

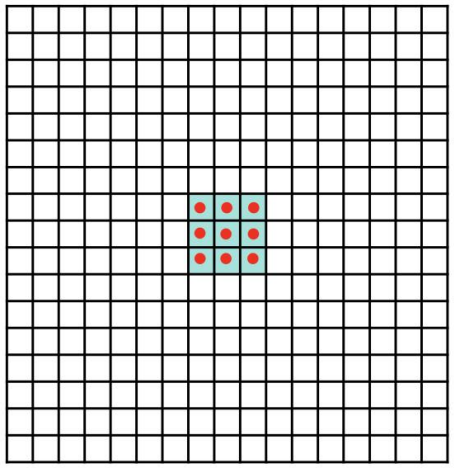
Add P circles of zeros outside of the original map

$$\text{height}_{new} = \lceil (\text{height-filter+2*pad}) / \text{stride} \rceil + 1$$

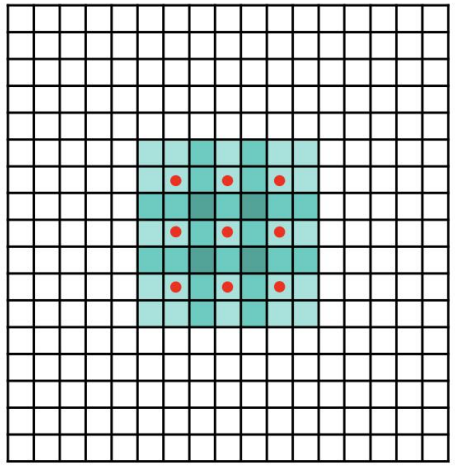
$$\text{width}_{new} = \lceil (\text{width-filter+2*pad}) / \text{stride} \rceil + 1$$

Dilated Convolution

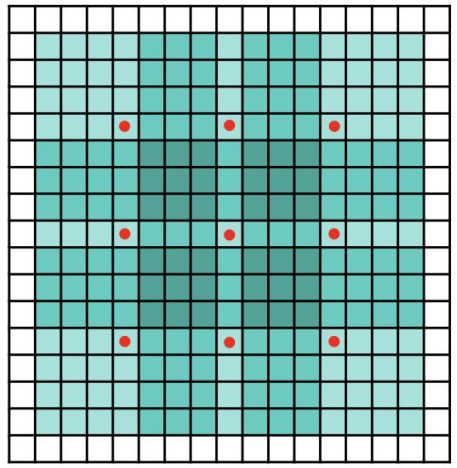
- Support exponential expansion of the receptive field without loss of resolution or coverage



3x3 1-dilated conv



3x3 2-dilated conv



3x3 4-dilated conv

Convolution: Formulation

- When working with images, we usually think of the input I and output Z of convolution as being 3D tensors:

$$Z_{j,k}^{(i)} = \sum_{l,m,n} I_{j+m-1,k+n-1}^{(l)} K_{m,n}^{(i,l)} + b^{(i)}$$

  - $Z_{j,k}^{(i)}$ is the value of output unit within channel i at row j and column k
  - kernel k is a 4D tensor with element $K_{m,n}^{(i,l)}$ giving the connection strength between an output unit in channel i and an input unit in channel l, with an offset of m rows and n columns between them.
- We may skip over some positions of the kernel in order to reduce the computational cost:

$$Z_{j,k}^{(i)} = \sum_{l,m,n} I_{(j-1)\times s+m,(k-1)\times s+n}^{(l)} K_{m,n}^{(i,l)} + b^{(i)}$$

  - s is the stride of this downsampled convolution
- Suppose we want to minimize some loss function $J(K, b)$

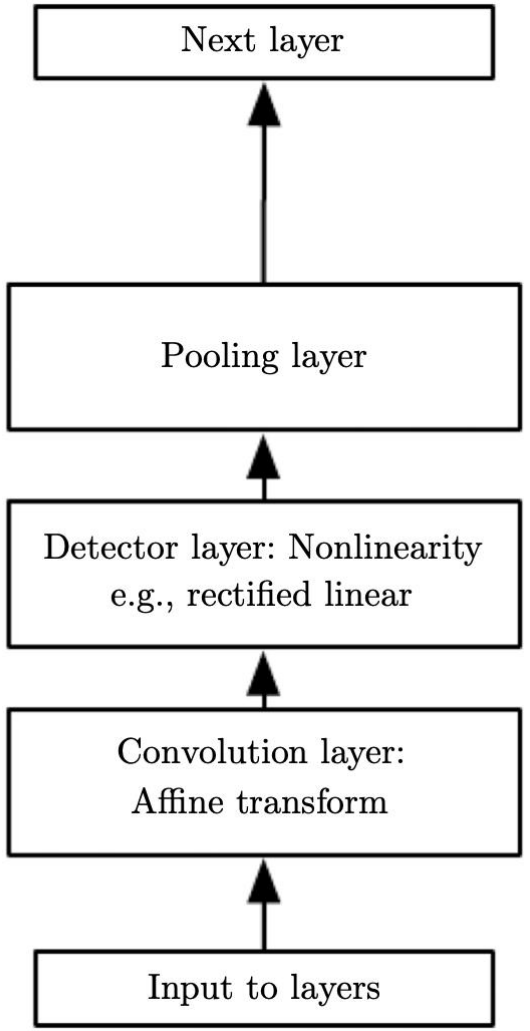
Pooling

- A convolutional layer consists of several layers (stages)
  - Convolution layer (stage)

$$Z_{j,k}^{(i)} = \sum_{l,m,n} I_{j+m-1,k+n-1}^{(l)} K_{m,n}^{(i,l)}$$

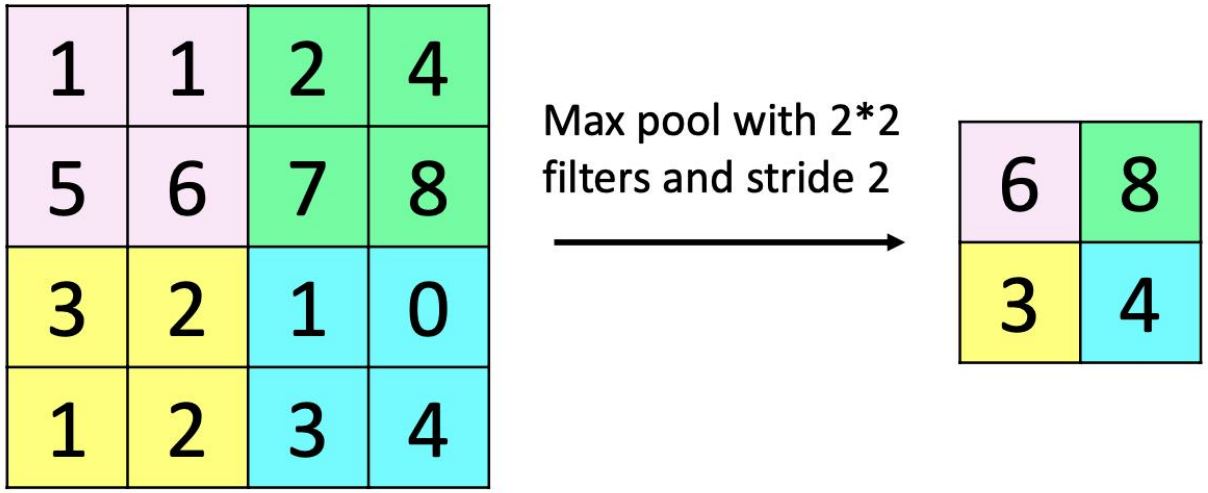
  - Nonlinerity laye
  - Pooling layer (stag

$$O = f(\sum_i X_i K_i + b)$$



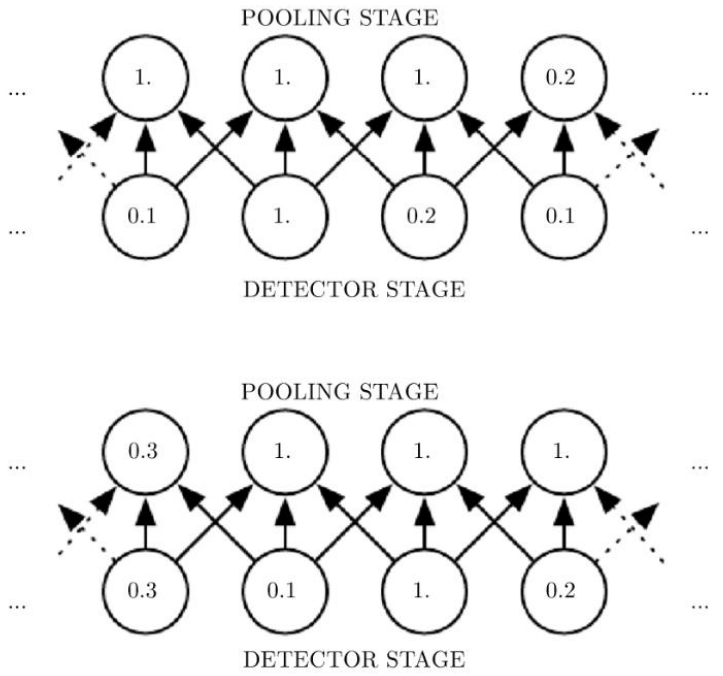
Pooling

- A pooling function replaces the output of the layer at a certain location with a summary statistic of the nearby outputs.
  - Max pooling
  - Average pooling
  - L2-norm pooling
  - Probability weighted pooling



Pooling

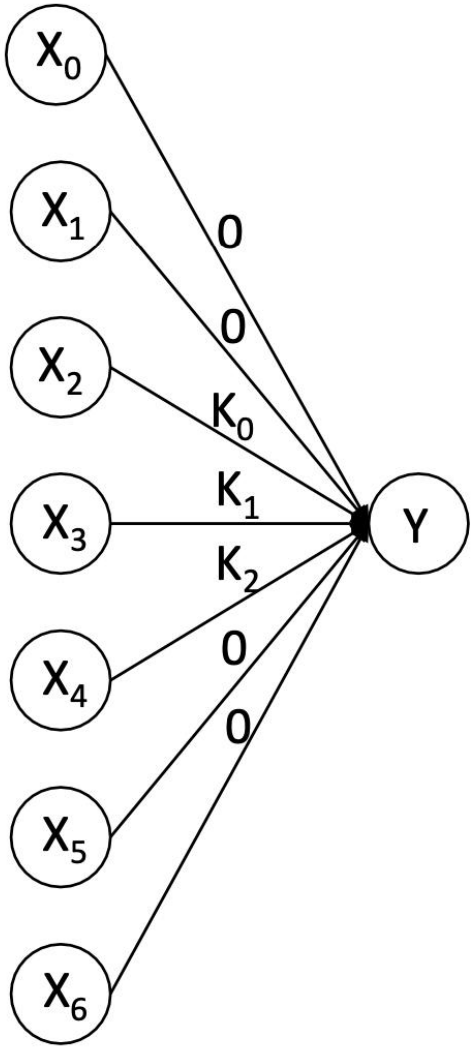
- Properties of pooling:
  - invariance to small translations of the input
  - improve statistical efficiency and reduce memory requirements
  - reduce the input of next layer handle inputs of variable size



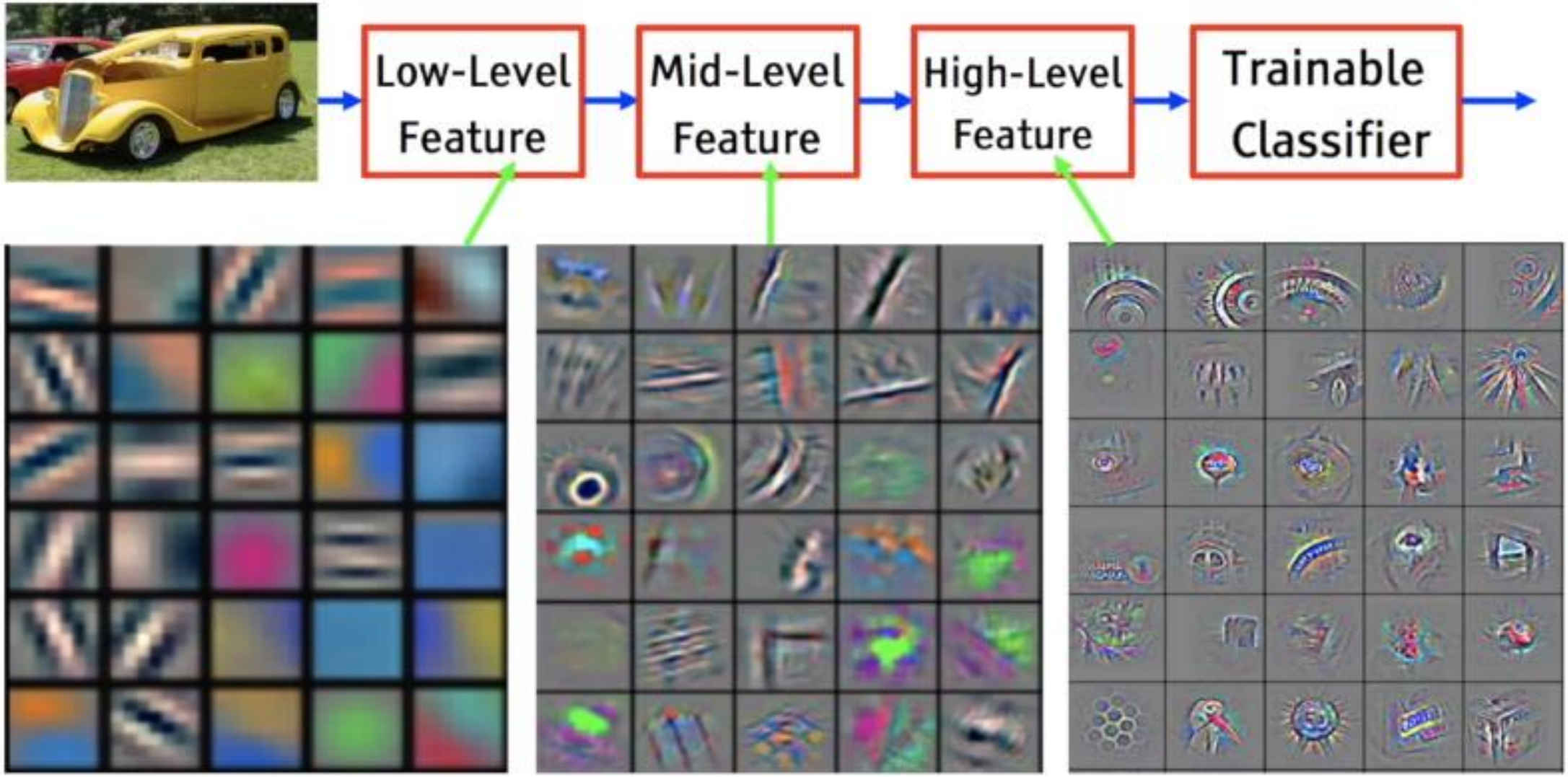
Max pooling introduces invariance

Convolution and pooling as an infinite strong prior

- A convolutional layer can be viewed as a fully-connected layer with an infinitely strong prior over its weights.
- This prior is that the weights for one hidden neuron must be identical to the weights of its neighbor, but shifted in space.
- In other words, the weights must be zero, except for in the small field assigned to that hidden neuron.
- Likewise, the use of pooling is an infinitely strong prior that each unit should be invariant to small translations.

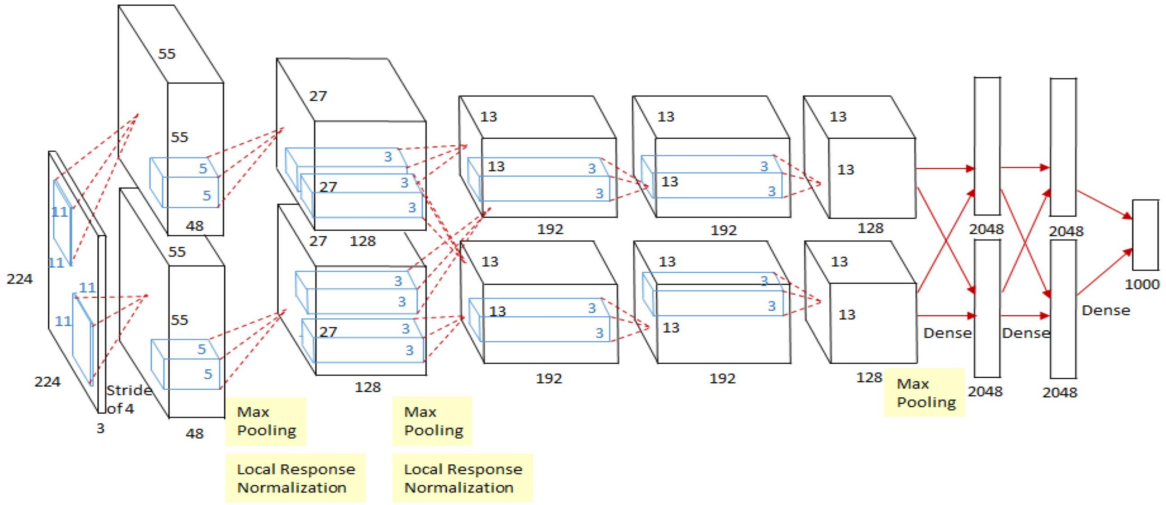


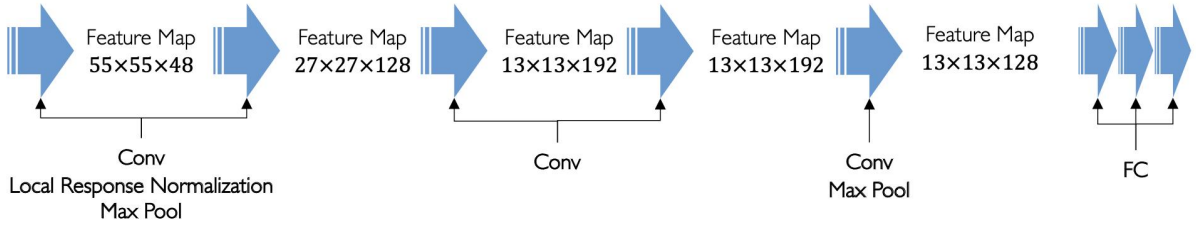
Feature detected by CNNs



AlexNet 2012

- The earliest representative “deep learning” model is the AlexNet
- AlexNet is the first deep CNN method that achieves significant improvement in image classification.





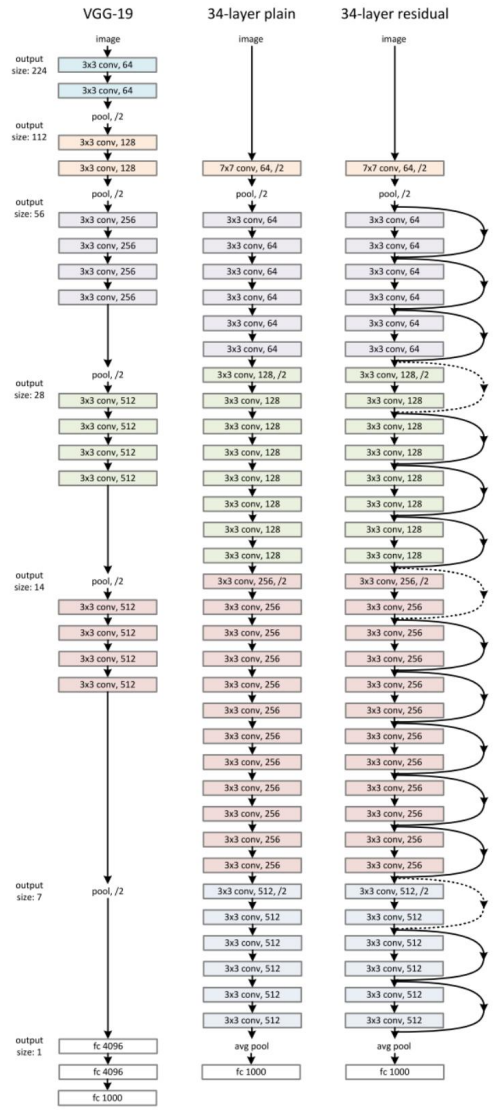
- AlexNet uses a deep CNN with 7 layers, ReLU, and dropout.
- This great success innovates the development of CNN.

Deeper CNNs — ResNet 2015

- MSRA uses the ensemble of ResNet, and wins most of the competitions ILSVRC 2015.

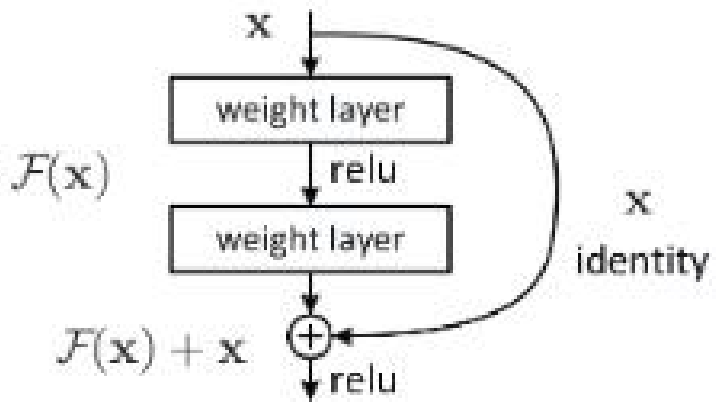
- Right illustrates ResNet-34 architecture (compared with VGG-19 and CNN-34)

- ResNet uses residual connection, therefore the network can reach as much as 1,202 layers while still gaining performance.



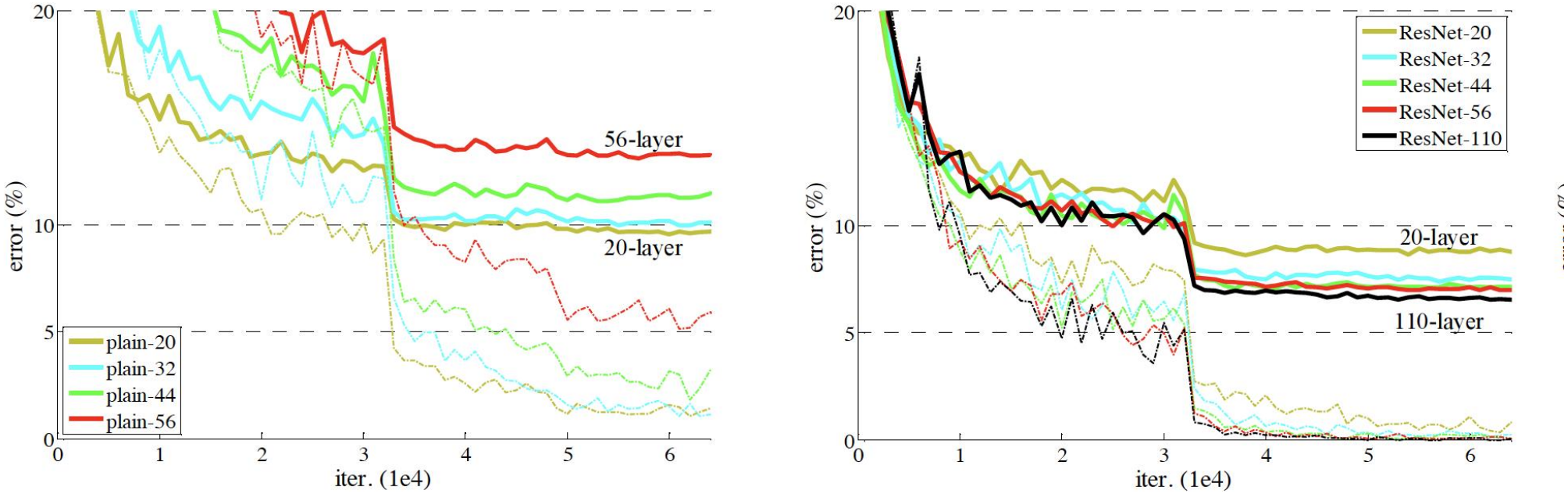
Residual Neural Networks

- Residual Neural Networks (He et al. 2015) **sums up** low level hidden representation and high level hidden representation in a neural network.
- The high level representation tries to learn the **residual** between the low level representations and the targeted labeling manifold.



A Residual block in ResNet.

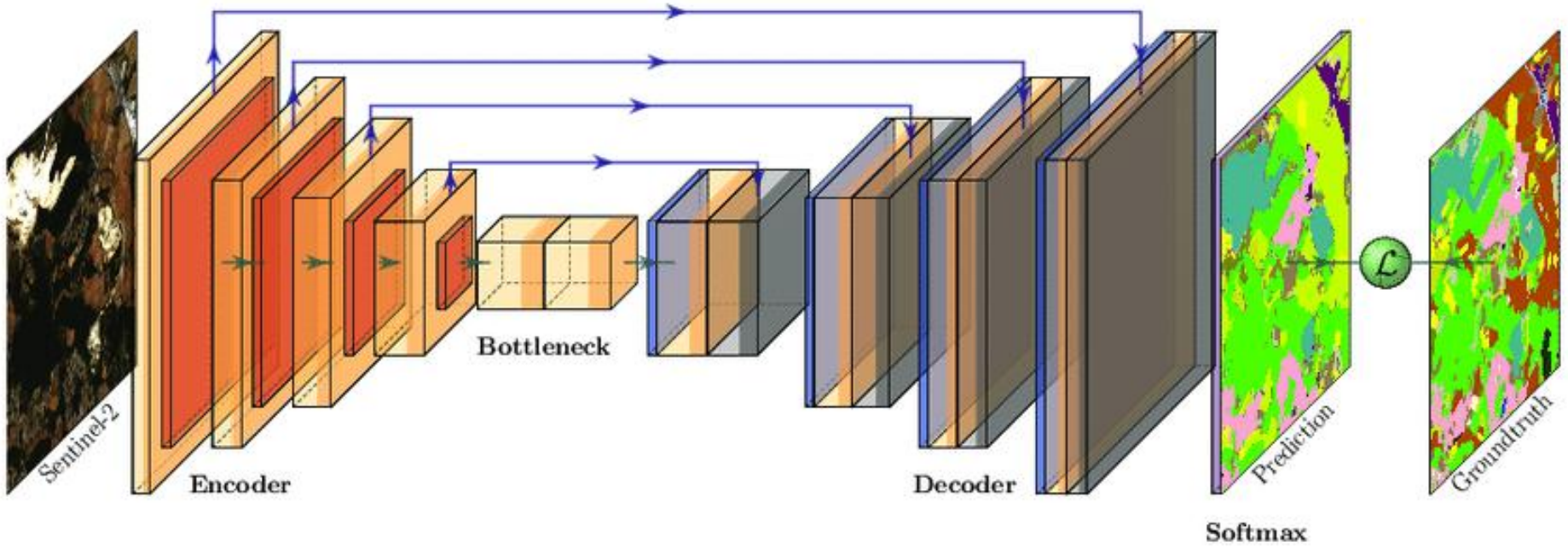
Residual Neural Networks



Error rate for plain CNN and Resnet.

- Residual Neural Networks performs slightly better than normal CNNs with the same depth.
- Residual Neural Network structure learns the residual while keeping lower level representations, thus more stable to be expanded **very deeply**.

U-Net 2015

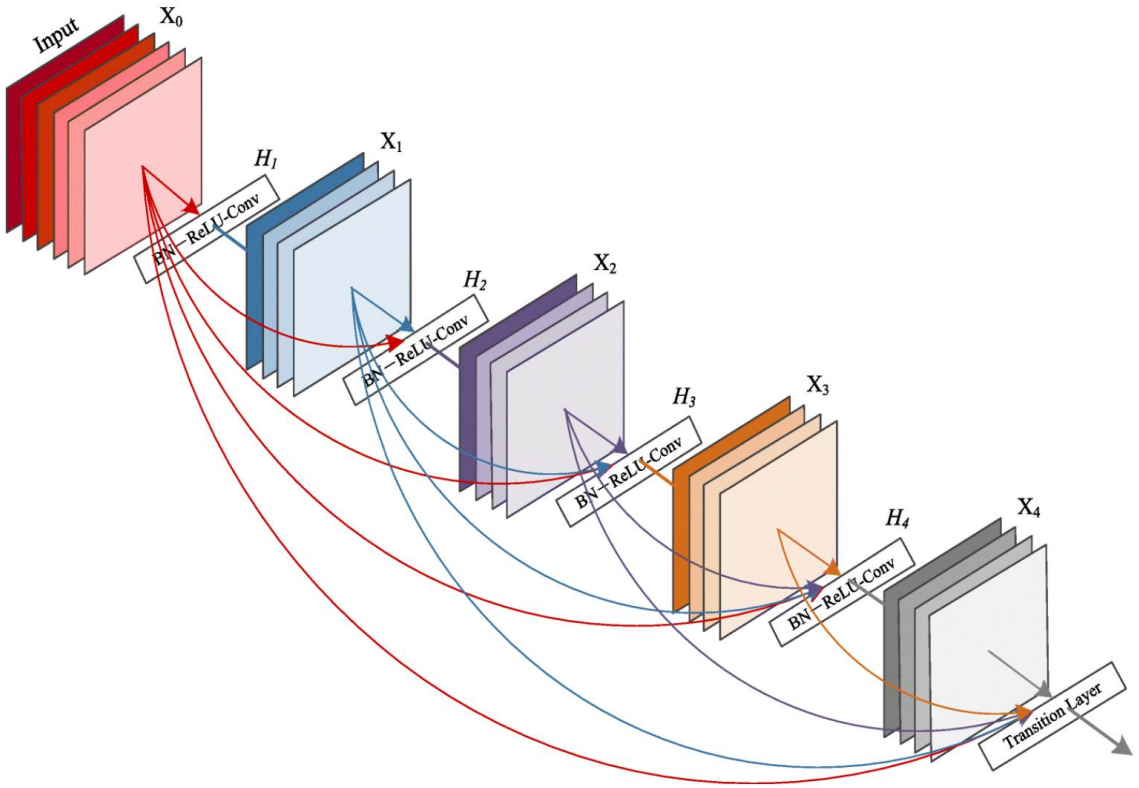


- It was first introduced by **Olaf Ronneberger** et al. , primarily for **biomedical image segmentation**. It can also be used in any network that needs to generate 2d information, such as **image generation**, **style transfer**, **depth estimation**, etc.

- **U-shaped Structure**: Symmetrical encoder (contracting path) and decoder (expanding path).

- **Skip Connections**: Connects symmetrical layers between the encoder and decoder to preserve detailed information and improve feature transfer.

Densely Connected Neural Networks 2017



- Every layer is connected with all lower level layers, instead of only the previous one.
- It is also able to be expanded very deeply, since it also keeps the low level representations.

Densely Connected Neural Networks

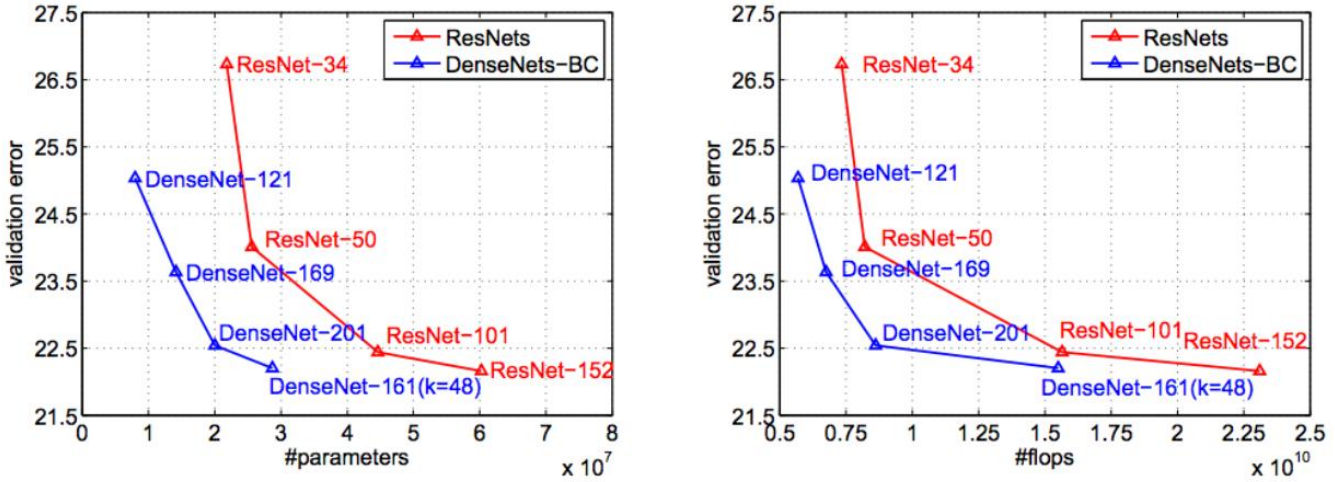


Figure 3. Comparison of the DenseNet and ResNet Top-1 (single model and single-crop) error rates on the ImageNet classification dataset as a function of learned parameters (*left*) and flops during test-time (*right*).

- DenseNet performs better than ResNet using same number of parameters/ same amount of flops.
- However, the densely connected property leads to less parallel computational efficiency.

A Tradeoff!

Thank you!

Jie Tang, KEG, Tsinghua U

http://keg.cs.tsinghua.edu.cn/jietang
https://github.com/THUDM