4-1 Convolution

Zhonglei Wang

WISE and SOE, XMU, 2025

Contents

Intuition for convolution neural network

1. Given an image, why not use a fully connected neural network?
    - Number of parameters is overwhelmingly large, easily leading to overfitting
        - ▷ To overcome this problem, we may require more training data, more memory and faster computation algorithms
        - ▷ Almost impossible in practice
    - Overlooks the spatial structure
    - Not location (translation) invariant
2. Consider a new network, convolution neural network, for image processing

Convolution neural network

1. Advantages of convolution:
   - Convolution is implemented for local regions independently, so computation can be parallelized
   - Convolution shares kernels over different local regions, so it involves fewer parameters
   - Convolution takes local spatial information into consideration, so the model is more spatially in
2. Neural network using convolution is called convolution neural network (CNN)

Convolution

1. Convolution is not new to us
   - In probability, convolution is related to the density function of $Z = X + Y$

$$f_Z(z) = \int f_X(x) f_Y(z - x) \mu(\mathrm{d}x),$$

     - ▷ μ : dominating measure
     - ▷ $f_\star(\cdot)$: density function for random variable $\star$
     - ▷ X, Y : two independent random variables
2. For simplicity, denote the convolution of functions f and g to be

$$h(z) = (f * g)(z)$$

   - Mathematically, f and g can be any functions as long as the definition is valid

1d convolution

Input	3	6	5	4	8	9	1	7	9	10	10
Kernel									-1	0	1

Calculation

Result	2	-2	3	5	-7	-2	8	3	1

1. Usually, we flip $g(x)$ to obtain a kernel so that we can conveniently use a command like `np.sum`
2. Values for x are no longer useful

Padding

1. Intuition
    - The size of the input sequence decreases after convolution
    - Some (important) information on the boundary is overlooked

Padding

1. Intuition
    - The size of the input sequence decreases after convolution
    - Some (important) information on the boundary is overlooked

Padding

Input	0	3	6	5	4	8	9	1	7	9	10	10	0
Kernel											-1	0	1
Calculation													
Result		6	2	-2	3	5	-7	-2	8	3	1	-10	

Padding

1. Two types of convolutions:
   - Valid convolution: no padding
   - Same convolution: apply padding so that the output size is the same as the input
2. We can pad more than one 0 around the boarders.
3. We can move more than one steps to do convolution to
   - Decrease the output size
   - Balance accuracy and computation cost

Stride

Input

3	6	5	4	8	9	1	7	9	10	10

Kernel

-1	0	1

Calculation

Result

2	3	-7	8	1

1. In the above example, stride is 2
2. Stride can be other positive numbers
3. Padding and stride can be used at the same time

Dilated/Atrous convolution

Input

3	6	5	4	8	9	1	7	9	10	10

Dilated Kernel

-1	0	1	0	1

Calculation

Result

10	7	4	12	2	8	18

1. In the above example, dilation rate is 2
2. Dilation increases the receptive field without (essentially) increasing kernel size
3. Padding, stride and dilation can be used at the same time

2d convolution

Original sequence
(Padding = 1)

0	0	0	0	0	0	0
0	3	6	5	4	8	0
0	9	1	7	9	6	0
0	8	0	5	0	9	0
0	6	2	0	5	2	0
0	6	3	7	0	9	0
0	0	0	0	0	0	0

Kernel

-1	1
-1	1

Result
(Stride=2)

3	-1	4
17	11	6
12	2	6

Average kernel

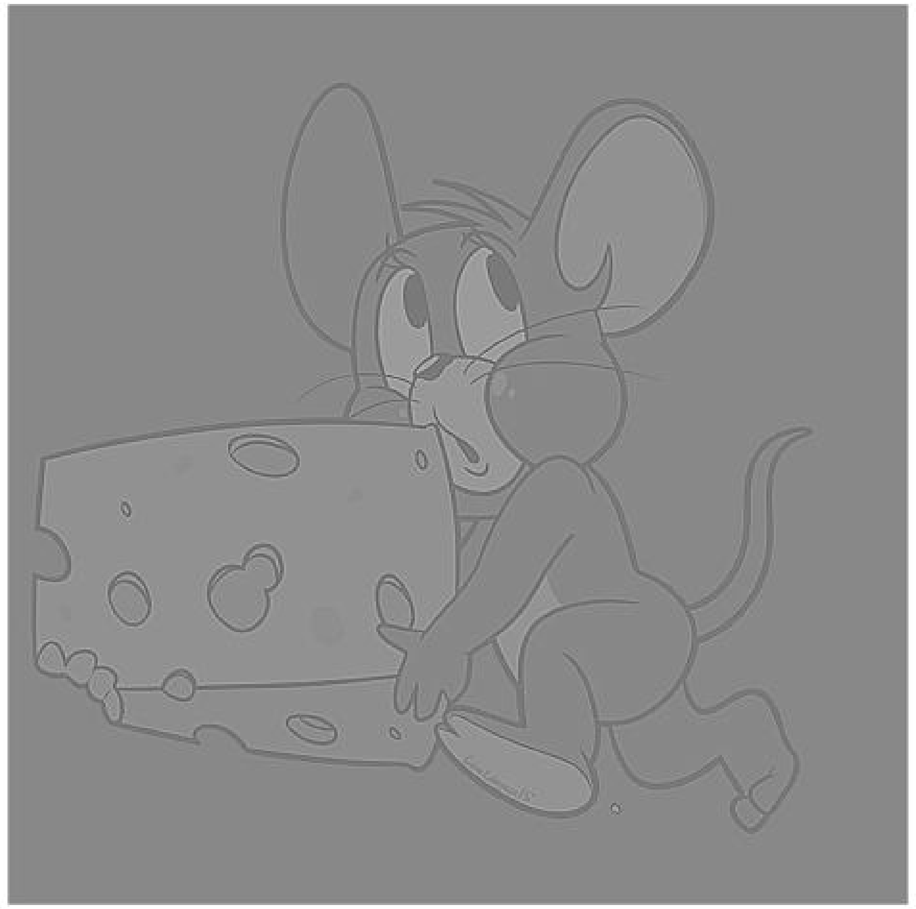
$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

Gaussian kernel

$$\begin{bmatrix} 1/16 & 1/8 & 1/16 \\ 1/8 & 1/4 & 1/8 \\ 1/16 & 1/8 & 1/16 \end{bmatrix}$$

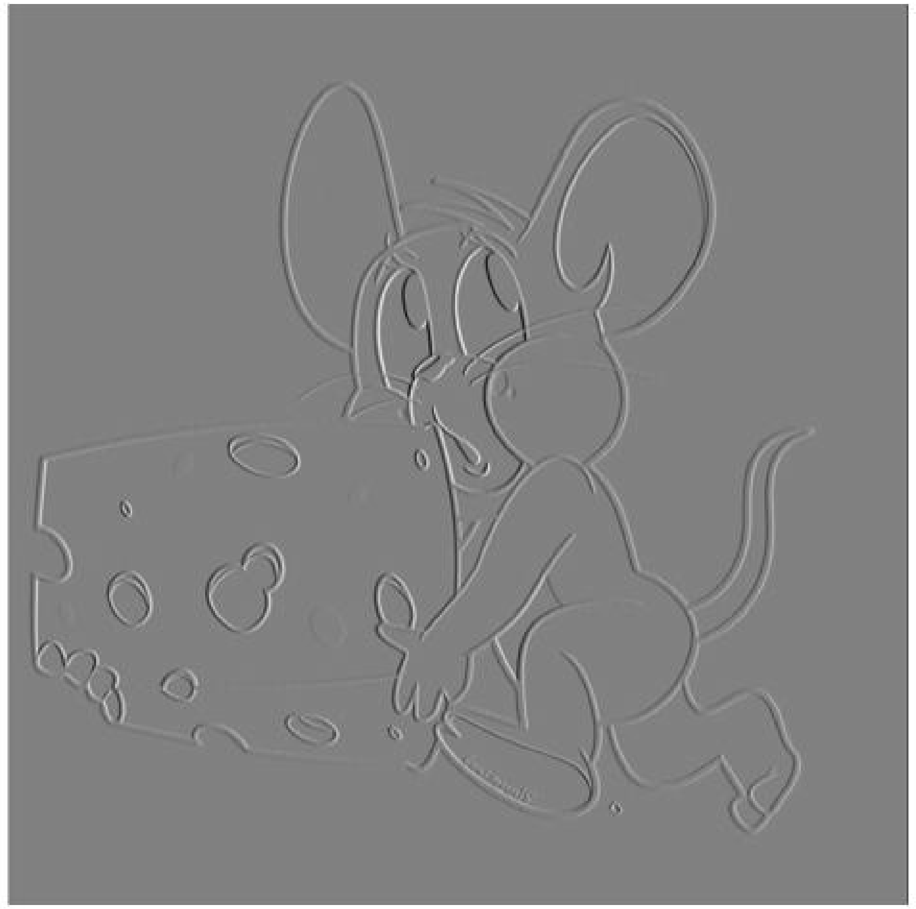
Sharpen kernel

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$



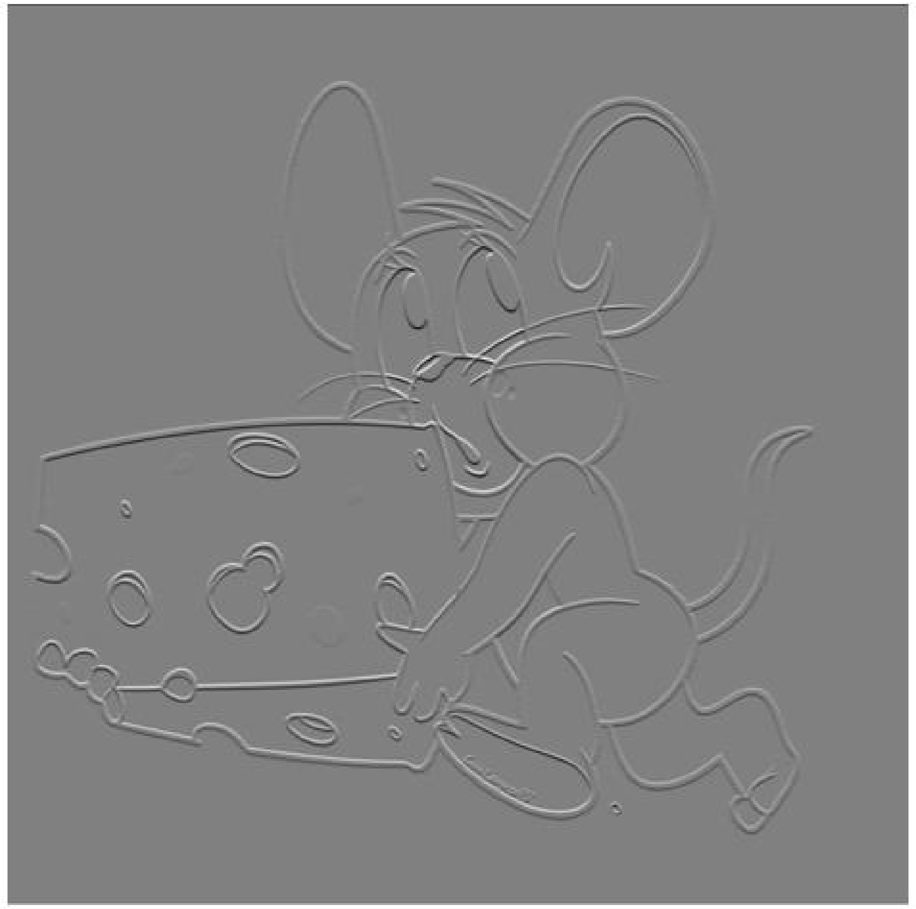
Sobel kernel for vertical boundary

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$



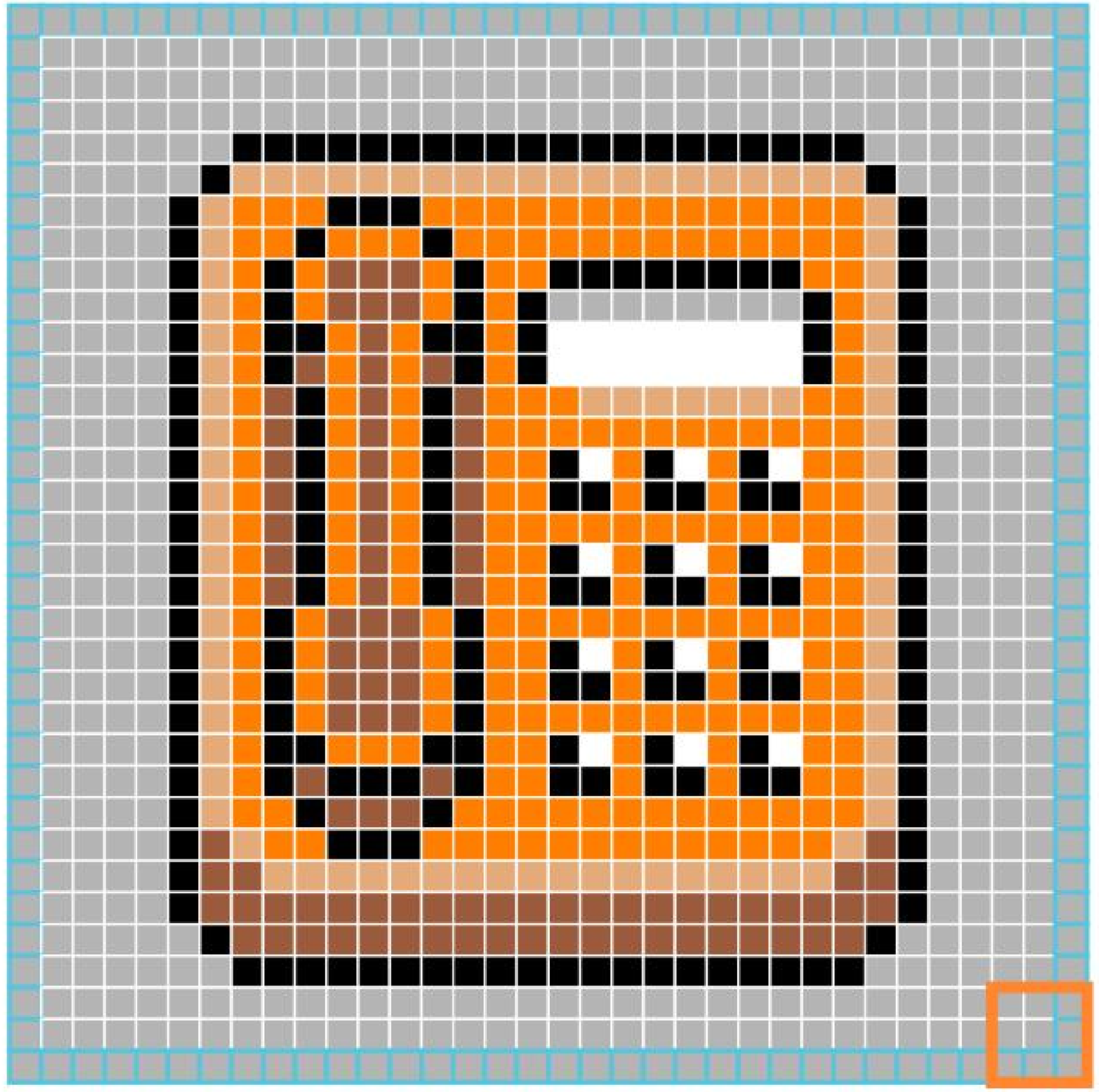
Sobel kernel for horizontal boundary

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \longrightarrow$$



Another example

Original image



Kernel

−1	−1	−1
−1	9	−1
−1	−1	−1

Sharpen

−1	0	1
−2	0	2
−1	0	1

Sobel (vertical)

−1	−2	−1
0	0	0
1	2	1

Sobel (horizontal)

Multi-Channel convolution

1. In general, the third dimension of kernels is the same as the input
2. Output size calculation
   - Input size: $d_H \times d_W \times d_C$
   - Kernel size: $f \times f \times d_C$
   - Padding: p
   - Stride: s
   - Output size: $d'_H \times d'_W$

$$d'_H = \left\lfloor \frac{d_H + 2p - f}{s} + 1 \right\rfloor, \quad d'_W = \left\lfloor \frac{d_W + 2p - f}{s} + 1 \right\rfloor$$